**DAHLGREN DIVISION**
**NAVAL SURFACE WARFARE CENTER**
Dahlgren, Virginia 22448-5100

**NSWCDD/TR-94/285**

# REAL-TIME BIAS ESTIMATION AND ALIGNMENT OF TWO ASYNCHRONOUS SENSORS FOR TRACK ASSOCIATION AND FUSION

**BY JEFFREY E. CONTE AND RONALD E. HELMICK**

**SYSTEMS RESEARCH AND TECHNOLOGY DEPARTMENT**



**APRIL 1995**

## FOREWORD

The simulation model and analysis discussed in this report were developed by the Digital Systems Branch (B32) of the Combat Systems Technology Group (B30). The work was funded by the Independent Research Program and the Seed and Venture Program at the Naval Surface Warfare Center Dahlgren Division (NSWCDD). This report has been reviewed at NSWCDD by Rich Stutler, Head, Digital Systems Branch and R. Neal Cain, Head, Combat Systems Technology Group.

Approved by:

D. B. Colby

DAVID B. COLBY, Head
Systems Research and Technology Department

## ACKNOWLEDGEMENTS

The authors would like to acknowledge Dr. Theodore R. Rice for his helpful consultations and suggestions for practical examples. They would also like to acknowledge and thank Scott A. Hoffman and William D. Blair for their assistance in coding of one-step, fixed-lag smoothing algorithms. The authors would also like to acknowledge Gregory A. Watson for providing valuable consultation regarding filtering and for providing specialized target generation programs. We would also like to thank John Miniuk (F21) and Larry Sumner (F21) for providing the real data that is analyzed in Section 6.

## ABSTRACT

An extensive simulation study of the problem of relatively aligning two sensors that measure range, bearing, and elevation is described in this report. Simple simulations are used to demonstrate the effects of alignment errors on multisensor tracking. The theory and algorithms for relatively aligning the sensors are briefly summarized. In the original derivation, it was assumed that the data from the two sensors are synchronous. This work presents the extension and simulation of the algorithms to permit the use of asynchronous data. This is accomplished by using Kalman-filter-based prediction algorithms to time-align the state estimates from the two sensors. One-step, fixed-lag smoothing is also employed to improve the accuracy of the state estimates. The effectiveness of using the Interactive Multiple Model algorithm versus single model filtering in the tracking filters prior to bias estimation is also studied. Multiple model versions for prediction and one-step, fixed-lag smoothing of the track estimates are also applied and compared with their single model counterparts with respect to bias estimation accuracy. The bias estimation algorithm as described in this report requires time-aligned state estimates from the tracking filters of each sensor, or measurements from each of the two sensors, and uses Kalman filtering to estimate the sensor bias parameters, which are assumed to be constants. A simulation that illustrates the effect of sensor misalignment on the ability to accurately associate and fuse tracks in a multisensor system is described. The results of that experiment indicated that the percent of correctly associated tracks was between 90 and 95 percent when the sensors were properly aligned; and when the sensors were unaligned, the tracks were never associated. Finally, an example of aligning real data from a three-dimensional radar and a two-dimensional Electronic Support Measure (ESM) sensor measuring bearing and elevation is described. To accomplish the alignment, the heuristic of assigning the radar range to the ESM state is successfully used. It is concluded that the algorithm would prove to be useful for relatively aligning data from diverse tracking sensors that are stabilized and mounted near to each other, such as those found on a ship. This would increase the efficiency of a multisensor tracking system, lessen the tracking burden by providing more accurate data for track association and fusion, and provide a more accurate picture of the environment.

## CONTENTS

# ILLUSTRATIONS

# 1.0 INTRODUCTION

In an integrated multisensor tracking system, individual sensor tracks on mutual targets are fused into a single system track. This reduces the work load of the system if it is done accurately and efficiently. One of the most challenging problems encountered when attempting to produce accurate fused tracks is that of aligning the data from the multiple sensors. Alignment is the process of expressing the data from the sensors in a common reference frame, where errors in the transformation process are minimized. Alignment is particularly important in the integration of stand-alone sensors into multisensor systems to provide performance improvements and enhanced capabilities for tracking and surveillance. The presence of uncompensated alignment errors will degrade the overall performance of multisensor systems; for example, they can produce redundant tracks on the same target and they can lead to composite (or system) tracks that are less accurate than the local tracks from the individual sensors.

One source of alignment errors is calibration errors (i.e., offsets) in the measurements provided by the sensors. Although the sensors are usually calibrated in an initial calibration procedure, the calibration may deteriorate over time. Another source of registration errors is attitude (or orientation) errors in the reference frames of the sensors. Most sensors incorporate some technique for compensation of rotational movements in the reference frame of the sensor (usually a gyroscopic device to determine the attitude of the sensor). Tracking systems can be grouped into three major classes according to the type of compensation: (1) Unstabilized - the sensor operates in a totally unstable environment, the data from the sensor contains rotational components, and gyroscopic compensation is external to the sensor; (2) Partially stabilized - the sensor is stabilized along one axis, its data contain rotational components in the unstabilized axes, and the remainder of the compensation is external to the sensor; (3) Fully stabilized - the sensor is gimbal-mounted and its data contain no rotational components.[1] Attitude errors in the reference frame can be caused by bias errors in the gyros and/or gimbals. Other sources of errors include

sensor location errors caused by bias errors in the navigation systems associated with the sensors and timing errors in the clocks of the sensors.

Simulation studies that are described in this report have shown that, if alignment errors are not corrected and if they are large enough, a fused system track will actually be less accurate than the unfused tracks from the individual sensors, and they can even lead to an inability to associate the tracks from the two sensors. *Large enough* in this context can be, for example, a bias of just 0.6 deg in bearing alone. If such errors occur, the system may not be able to correlate the tracks from the individual sensors, and must therefore maintain separate tracks on a single target. This increases the processing burden and provides an inaccurate picture of the environment. On the other hand, if the biases can be estimated and removed, the fused track will most likely be superior to the individual tracks, and accurate track association can be maintained, thus reducing the system load and producing a more accurate picture of the environment. While biases may often be assumed to be constant over short times, periodic readjustment is usually needed in the long term. The algorithms discussed in this report enable on-line recursive bias estimation so that alignment can be adjusted in real-time or near real-time.

This study represents an extension and application of theory derived in references (2) and (3). The new work involves the comparison of the use of single-model versus multiple-model filtering algorithms and time-aligning asynchronous data using prediction alone versus prediction and one-step, fixed-lag smoothing. There are four facets to the work described in this report. First, simple simulations are used to demonstrate the effects of alignment errors on multisensor tracking. Secondly, a model is described that simulates an environment in which two, three-dimensional (3D) sensors track mutual targets in the presence of noise and with sensor biases in range, bearing, elevation, roll, pitch, and yaw. The bias estimation technique described in reference (2) is used, making use of one or two targets that are being tracked by both sensors, to recursively estimate the relative biases between the two sensors. The theory and method of bias estimation described in reference (2) permits alignment of two sensors relative to each other, rather than absolutely aligning them with respect to the truth. Relative alignment is sufficient for accurate track association, and, in fact, absolute alignment is not possible in this context because the truth is not observable by either sensor. To obtain true information for alignment, a third source of accurate position information, such as the Global Positioning System, would be required. In such a case,

the biases for each sensor would be estimated separately, and both tracks would be corrected to align with the truth. Due to simplifying assumptions that will be made, this theory applies when the sensor locations are accurately known, the biases are small, and the distance between the sensors is small enough to employ flat-earth transformations.

The third facet of this work is the comparison of various methods for improving bias estimation, such as tracking with multiple versus single model filters, and time aligning measurements by prediction and one-step, fixed-lag smoothing versus prediction alone. The theory assumes that the data from the sensors are synchronous, though in practice this is seldom the case. It will be seen that this assumption can be relaxed if accurate time-alignment techniques are used. The results of simulation studies used to make these comparisons will be presented. The method and results of associating the aligned and unaligned tracks will also be discussed. Finally, the theory will be applied to the alignment of real data from a two-dimensional (2D) Electronic Support Measure (ESM) sensor measuring bearing and elevation and a three-dimensional (3D) sensor measuring range, bearing, and elevation, by assigning the range from the 3D radar to the 2D ESM sensor. The studies presented in this report indicate that real-time relative sensor alignment is both feasible and practical for systems meeting the requirements imposed by the assumptions made in deriving the bias estimation algorithm, and for some systems that do not strictly meet those requirements.

The remainder of this report is organized as follows. Section 2.0 presents the effects of alignment errors on multisensor tracking. Section 3.0 presents background information regarding the theory for the relative alignment of 3D sensors and the model used for simulating the alignment process. In Section 4.0, the alignment simulation process is described. The process models tracking, aligning, and associating tracks from two misaligned, asynchronous sensors that are tracking mutual targets. Section 5.0 presents the results of studies performed using the alignment model. Section 6.0 gives the results of applying the technique to real data from a 2D ESM sensor measuring bearing and elevation only, and a 3D radar, using the heuristic technique of assigning the radar range to the 2D ESM sensor, and assuming no range bias. Finally, Section 7.0 summarizes the results of this effort.

# 2.0 EFFECTS OF ALIGNMENT ERRORS ON MULTISENSOR TRACKING

## 2.1 INTRODUCTION

In this section, simple simulations are used to demonstrate the effects of alignment errors on multisensor tracking. There are two basic levels at which multisensor tracking can be performed: (1) sensor-level tracking with track-to-track fusion, and (2) central-level tracking with measurement fusion. In sensor-level tracking with track-to-track fusion, each sensor maintains its own local track file, where the tracks in the local track files are established and maintained using raw measurements received from the local sensor only. The local track files are then sent to a fusion center where they are combined into central-level (or composite) track files. In central-level tracking with measurement fusion, each sensor reports its raw measurements directly to a central processor, where a central-level track file is established and maintained using the raw measurements. There are also several hybrid approaches which combine facets of both the track-to-track fusion and measurement fusion approaches.

The effects of alignment errors on both sensor-level tracking with track-to-track fusion and central-level tracking with measurement fusion are examined in this section. To simulate the effects of alignment errors on multisensor tracking, first a *true* target trajectory is simulated. This trajectory is for a constant velocity (CV) target that is closing and moving with a speed of Mach 1. This true trajectory is then passed to functions that simulate two 3D sensors, which simulate the sensors' measurements in spherical coordinates (range, bearing, and elevation). These sensor simulations introduce a bearing (or azimuth) bias into the measurements, and zero-mean Gaussian measurement noise is also added to the spherical measurements. For simplicity, only the effect of a bearing misalignment is examined. These simulated measurements are passed either to local trackers (i.e., sensor-level tracking with track-to-track fusion) or to a single centralized tracker (i.e., central-level tracking with measurement fusion).

## 2.2 SENSOR-LEVEL TRACKING WITH TRACK-TO-TRACK FUSION

In sensor-level tracking with track-to-track fusion, each sensor maintains its own local tracks, which are established and maintained using a local tracker that processes the raw measurements received from the local sensor only, and the local tracks are sent to a fusion center where they are associated with tracks from other sensors. Any associated tracks are then fused to produce an updated state estimate. The track-to-track association and fusion functions are described in more detail in Section 4.9. Briefly, the track-to-track association function is a statistical test that uses the difference in the track estimates reported by the two sensors to determine the *closeness* of the tracks. Those tracks that are *close* are assumed to be from the same physical target, and they are said to be associated. The track estimates from the two sensors for an associated track can be combined (or fused) using equations (4-63) and (4-64) to produce an estimate that *theoretically* is more accurate than the track estimates produced by either sensor.

In these simulations, two 3D sensors are tracking the CV target, where one of the two sensors is unbiased (i.e., no bearing bias) and only the other sensor has a bearing bias. The sensors provide position measurements of the target with a period of 1 sec, and the measurements reported by the sensors are synchronous. The local trackers at the two sensors are identical. The trackers are extended Kalman filters that are designed for linear CV dynamics with piecewise constant acceleration process noise. Each tracker uses its current position measurement in spherical coordinates and its previous Cartesian state estimate and covariance to produce an updated Cartesian state estimate and covariance.

For the first simulation, the effect of the bearing bias on track-to-track association is examined. The bearing bias is held fixed, and the standard deviation in the accuracy of the sensors' bearing measurement is varied. The standard deviations in the range and elevation measurements are fixed at 25 m and 0.2 deg, respectively. In the simulation, 100 Monte Carlo runs were performed for each value of the bearing standard deviation. Figure 2-1 presents a family of curves for the probability of associating the two tracks reported by the sensors on the same target versus the noise standard deviation. Each curve represents a different fixed bearing bias. These results indicate that track-to-track association is most affected by a fixed bias when the standard deviation of the measurement noise is small with respect to the fixed bias. This occurs because the

association algorithm is based upon the difference between the tracks in the presence of Gaussian noise. When the noise standard deviation is small compared to the fixed bias, the bias causes the difference between the tracks to be larger than that for which the association threshold is designed, and the association test decides that the tracks are from different targets, even though they actually originate from the same physical target. However, if the noise standard deviation is comparable to the bias, the difference between the tracks is more likely to fall within the region of the distribution in which the tracks will be associated, and the degradation in performance is less severe.

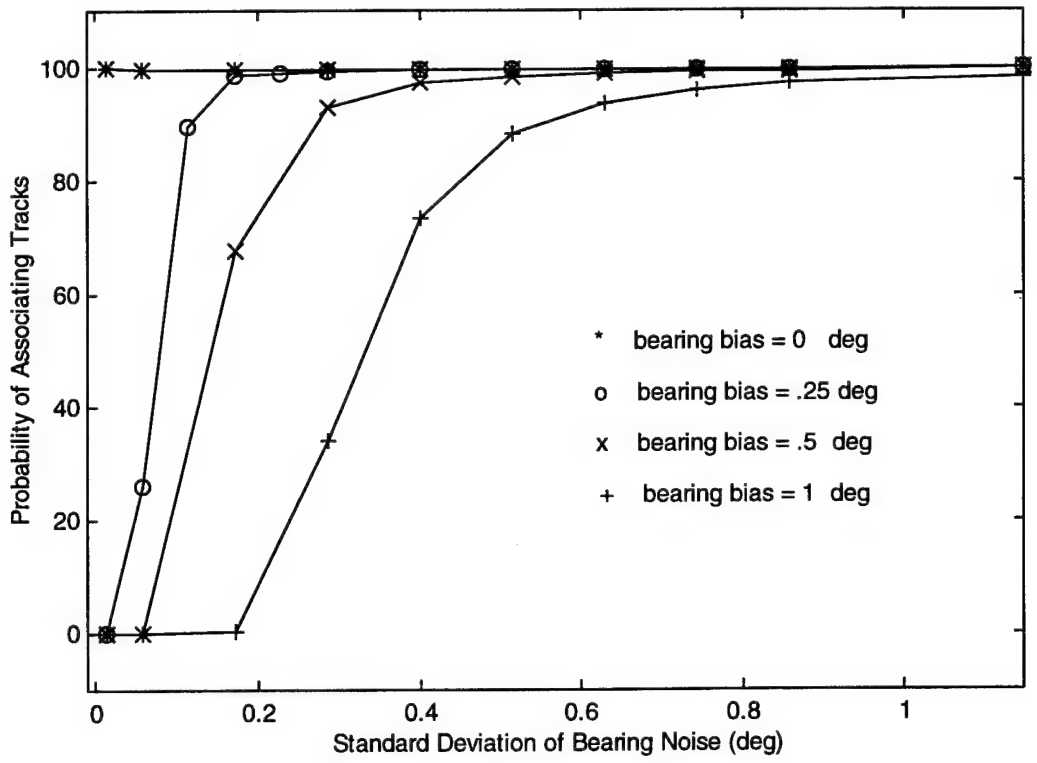


FIGURE 2-1. PROBABILITY OF ASSOCIATING TRACKS VS. STANDARD DEVIATION OF BEARING NOISE FOR FIXED BEARING BIASES

In the second simulation, the effect of the bearing bias on the accuracy of the fused state estimate is examined. For simplicity, it will be assumed that the two tracks reported by the sensors for the same target are associated; that is, no track-to-track association test is used. The two tracks

are fused using equations (4-63) and (4-64). Again, the bearing bias is held fixed, and the standard deviation in the accuracy of the sensors' bearing measurement is varied. In this simulation, 100 Monte Carlo runs were performed for each value of the bearing standard deviation. The results are presented in Figures 2-2 through 2-5, where the average Root-Mean-Square-Error (RMSE) in position is shown for the separate tracks from each sensor and also for the fused track. In Figure 2-2, the bearing bias was fixed at 0 deg, and it can be seen that the individual tracks are nearly aligned, because their RMSEs in position are almost identical. The fused track RMSE is lower than the individual sensor RMSEs, which indicates that fusing the tracks is advantageous when the sensors are unbiased, which is an obvious result. If the sensor 2 bearing bias is increased to 0.25 deg, as in Figure 2-3, the RMSE of the second sensor begins to increase as compared to that of the unbiased first sensor. However, the fused track RMSE remains below that of both individual sensors, except when the bearing standard deviation is less than about 1 mrad (.06 deg), so that it is seen to be advantageous to fuse the tracks even with a slight uncorrected bias. When the bearing bias of sensor 2 is increased to 0.5 deg, it can be seen in Figure 2-4 that the fused track RMSE is larger than that of the individual sensor RMSEs, unless the bearing standard deviation is larger than about 5 mrad (0.29 deg). The trend continues as the sensor 2 bearing bias is increased to 1 deg, as seen in Figure 2-5, for which the fused track RMSE is lower than the individual track RMSEs only when the bearing standard deviation is greater than about 11 mrad (0.6 deg). One must keep in mind that there is no alignment in this experiment. The trend indicates that it can be advantageous to fuse unaligned tracks if the sensor bias is small, or if it begins to approach the magnitude of the random noise, but, for a given bias, there is a limiting noise magnitude below which fusing tracks do not improve the overall multisensor system performance.

The results of these simulations indicate that for sensor-level tracking with track-to-track fusion, the presence of uncompensated alignment errors can seriously degrade the overall multisensor system performance, and they may actually produce a performance that is worse than that obtained using only a single sensor. The amount of degradation depends on the magnitudes of the alignment errors and the magnitudes of the random noises present in the system. For example, it may still be advantageous to employ sensor-level tracking with track-to-track fusion in the presence of uncompensated alignment errors, if the magnitudes of the alignment errors are of the same order or smaller than the magnitudes of the random noises. Of course, the best performance

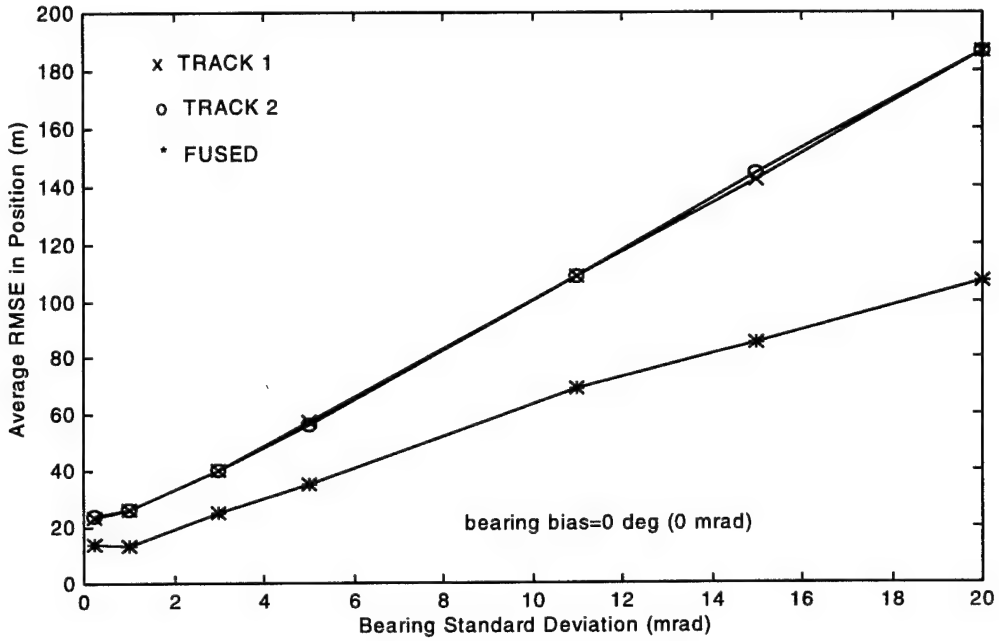


FIGURE 2-2. AVERAGE RMSE IN POSITION VS. BEARING STANDARD DEVIATION (BEARING BIAS = 0.0 DEG)

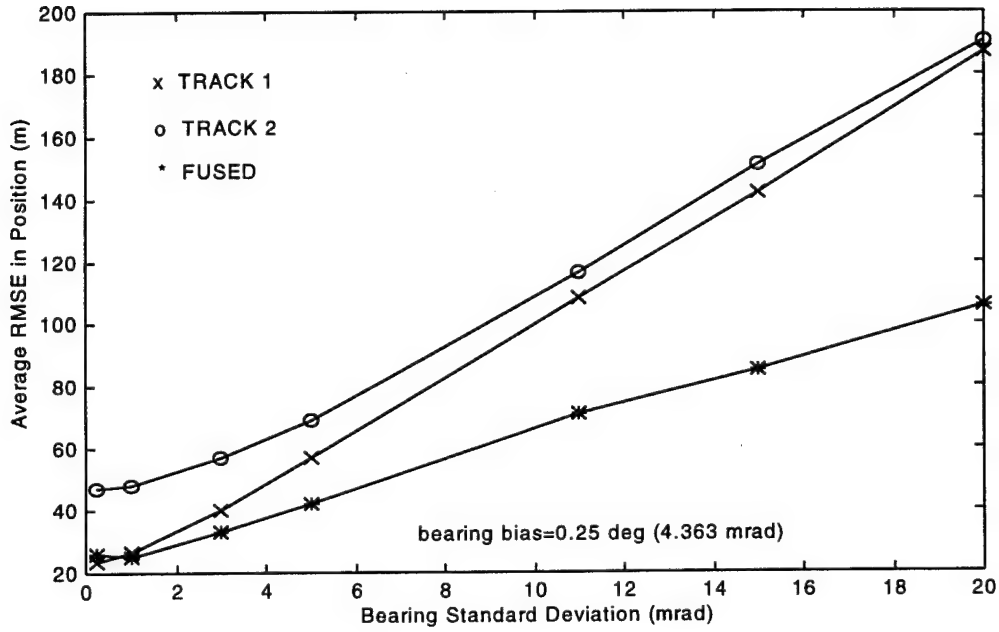


FIGURE 2-3. AVERAGE RMSE IN POSITION VS. BEARING STANDARD DEVIATION (BEARING BIAS = 0.25 DEG)

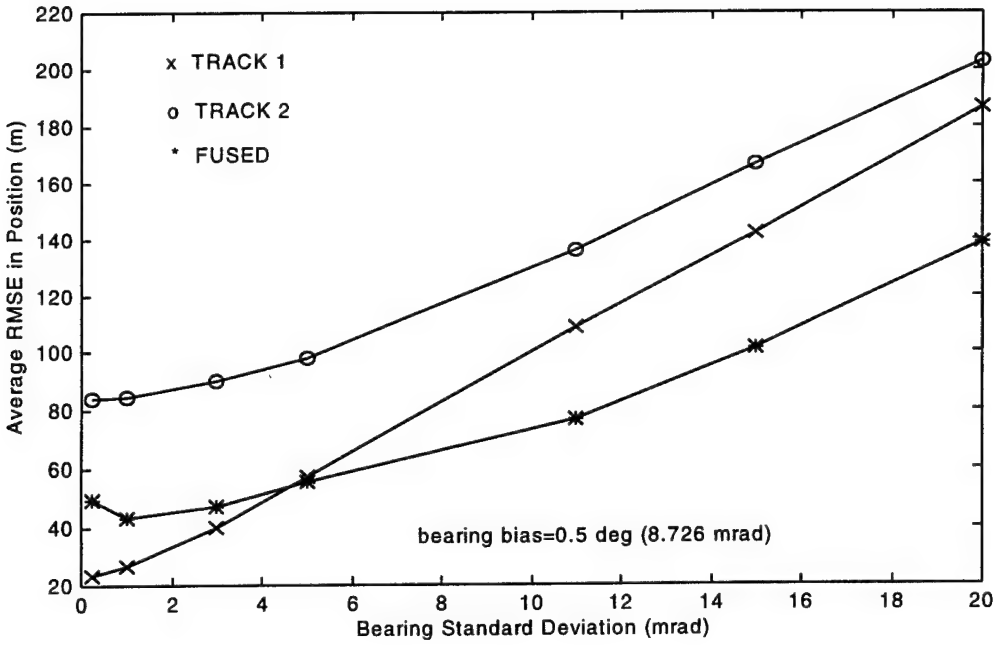


FIGURE 2-4. AVERAGE RMSE IN POSITION VS. BEARING STANDARD DEVIATION (BEARING BIAS = 0.5 DEG)

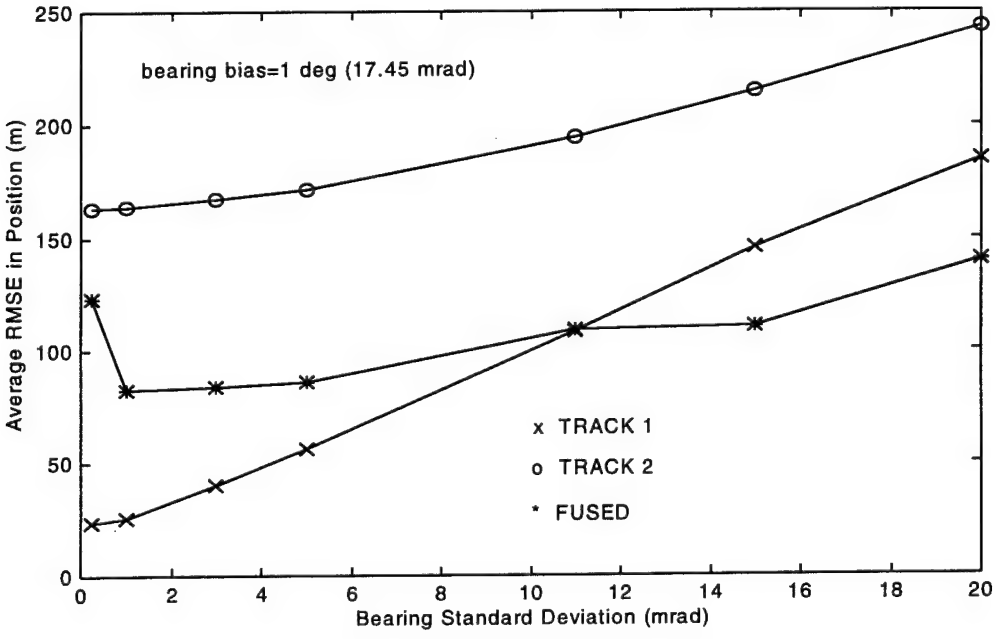


FIGURE 2-5. AVERAGE RMSE IN POSITION VS. BEARING STANDARD DEVIATION (BEARING BIAS = 1.0 DEG)

will be obtained if the presence of alignment errors can be detected, so that they can be estimated and then removed from the multisensor data if they are large enough to cause serious degradation in performance.

The RMSEs in the estimated velocity are not presented because the sensor bias had negligible effect on the RMSEs in the velocity. This occurs because the sensor bias is an additive offset in the measured bearing, the offset is constant, and each sensors' measurements are processed locally. Since the local filter basically uses a differencing technique to estimate the velocity, the effect of the constant bearing bias is effectively eliminated in the sensor's local estimate of velocity. The fused estimate of velocity is not biased because the local velocity estimates are not biased. Thus, the constant bearing bias has negligible effect on the velocity in sensor-level tracking with track-to-track fusion. However, we will see below that this is not the case for central-level tracking with measurement fusion because the multisensor measurements are processed centrally.

## 2.3 CENTRAL-LEVEL TRACKING WITH MEASUREMENT FUSION

In central-level tracking with measurement fusion, each sensor sends its raw measurements directly to a central processor, which establishes and maintains a central-level track file using the measurements. Measurements are associated to existing central-level tracks using contact-to-track association, which is basically a measurement validation (or gating) procedure. Any associated measurements are then used in a filter to update the existing track. Briefly, the contact-to-track association function is a statistical test that uses the difference between the raw measurement and the predicted measurement, which is obtained by predicting the existing central-level track to the time of the raw measurement, to determine the *closeness* of the raw and predicted measurements. If the measurement is *close* to the predicted measurement, the measurement is assumed to have originated from the track; otherwise, it belongs to a different track.

In these simulations, two, 3D sensors are reporting measurements on the CV target, where one of the sensors is unbiased (i.e., no bearing bias) and only the other sensor has a bearing bias. Each sensor reports measurements with period of 1 sec, but they are offset by 0.5 sec. Thus, the effective period for the measurements at the central-level processor is 0.5 sec. The central-level tracker is an extended Kalman filter that is designed for linear CV dynamics with piecewise constant acceleration process noise. The tracker uses the current position measurement in

spherical coordinates and the previous central-level Cartesian state estimate and covariance to produce an updated central-level Cartesian state estimate and error covariance. The central-level track is initialized using only the measurements from the unbiased sensor. Actually, the central-level track uses the first five measurements from the unbiased sensor before any of the measurements from the biased sensor are used to update the central-level track. This is done to ensure that the central-level track is fairly accurate and that there are no transient effects before any *biased* data is used to update the track.

The effect of the bearing bias on contact-to-track association is examined first. In the simulation, if the CV track does not have an associated measurement for three consecutive update periods, the CV track is defined as a lost track and it is terminated. The bearing bias is varied, and the standard deviations in the range, bearing, and elevation measurements are fixed at 25 m, 0.2 deg, and 0.2 deg, respectively. In the simulation, 100 Monte Carlo runs were performed for each value of the bearing bias. The performance degradation due to the bearing bias is manifested in a large percentage of lost tracks. This can be seen in Figure 2-6, which presents the percentage of lost tracks versus the bearing bias. It can be seen that no tracks are lost when there is no bearing bias. As the bias increases, the number of lost tracks increases until 100 percent of the tracks are lost. Obviously, a central-level fusion system with this kind of bias problem would be severely dysfunctional.

In the next simulation, the effect of the bearing bias on the accuracy of the central-level state estimate is examined. For simplicity, it will be assumed that the measurements reported by the sensors are associated to the track; that is, no contact-to-track association test is used. Again, the bearing bias is varied, and the standard deviations in the range, bearing, and elevation measurements are fixed at 25 m, 0.2 deg, and 0.2 deg, respectively. In the simulation, 100 Monte Carlo runs were performed for each value of the bearing bias. The results are presented in Figures 2-7 and 2-8, which show families of curves representing the RMSE in position and speed, respectively, as functions of time for various values of the bearing bias. It can be seen that for each individual bias, the RMSE periodically rises because of the bias error in the sensor 2 measurement, and falls with the arrival of the unbiased sensor 1 measurements. As the bias is increased, the magnitude of these fluctuations increases dramatically. Note that the problem is nonexistent when there is no bearing bias. The results of these simulations indicate that for central-level tracking with measurement fusion, the presence of uncompensated alignment errors can produce dramatic system inaccuracies and the inability to maintain track on a target.

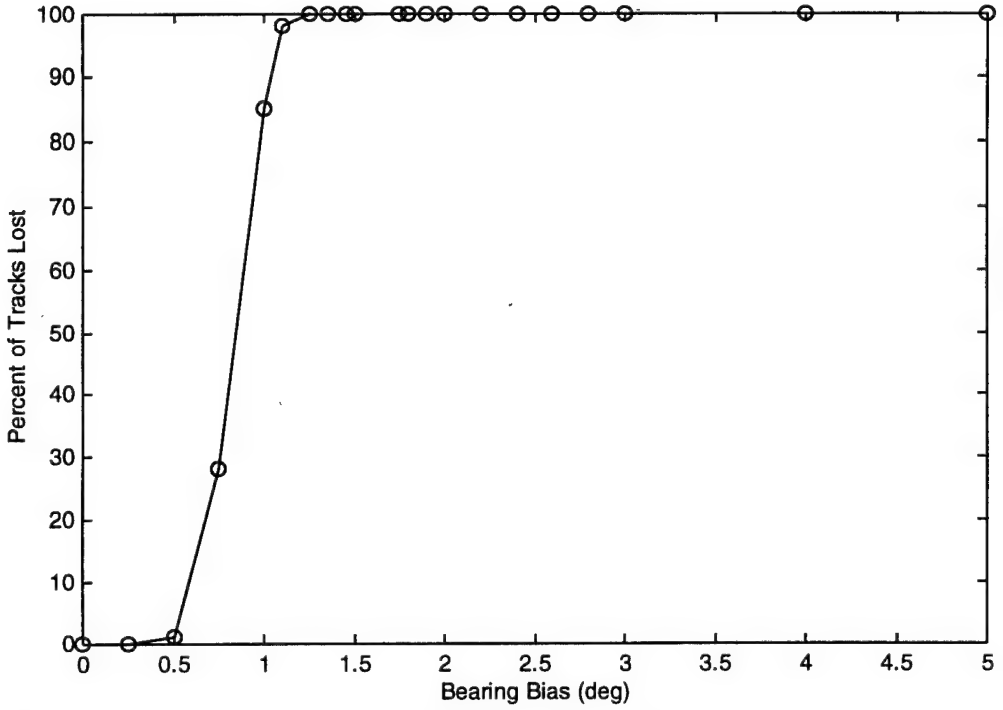


FIGURE 2-6. PERCENTAGE OF CENTRAL-LEVEL TRACKS LOST VS. BEARING BIAS

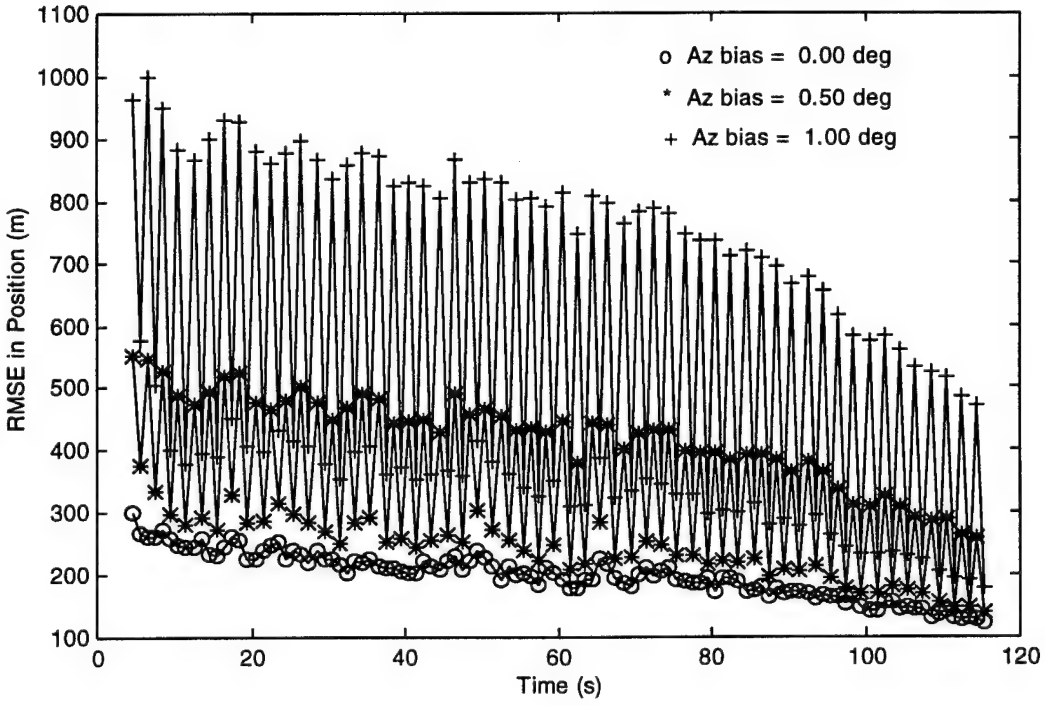


FIGURE 2-7. RMSE IN POSITION FOR VARIOUS BEARING BIASES

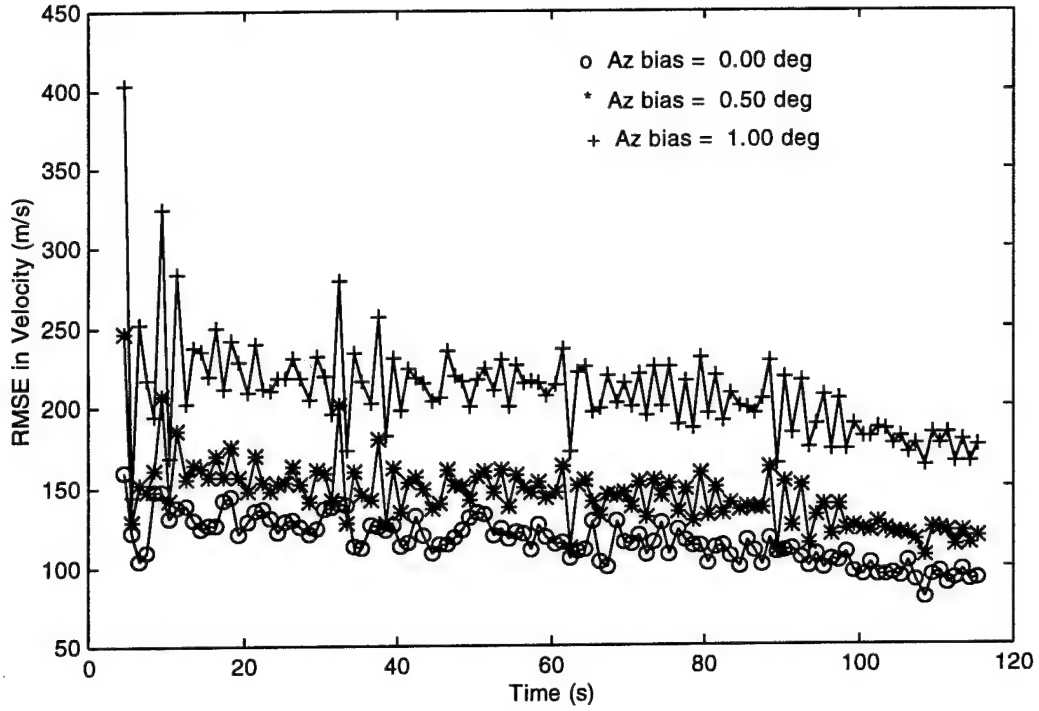


FIGURE 2-8. RMSE IN SPEED FOR VARIOUS BEARING BIASES

# 3.0 THEORY FOR RELATIVE ALIGNMENT OF 3D SENSORS

## 3.1 INTRODUCTION

The first step in solving the alignment problem for two 3D sensors is to decide which alignment errors are to be modeled. Two separate sources of alignment errors are included in the formulation of the problem in reference (2) - attitude errors in the reference frames of the sensors and offset errors in the measurements reported by the sensors. Sensor location errors are not included because the relative locations of sensors on a single platform can be accurately determined in an initial calibration procedure. Similarly, timing errors are not included because all of the sensors on a single platform can use a single centralized clock. Because of the method used to formulate the problem, it is not possible to determine the alignment errors in the individual sensors.[2] It is only possible to estimate the relative alignment errors between the sensors. This is sufficient to align one sensor relative to the other sensor; that is, one sensor is chosen as the *master* or *primary* sensor and the other sensor is aligned to it.

Below, the transformation between the reference frames of the sensors is considered first. This is followed by a presentation of the basic equations from reference (2) that are used to align the sensors. Finally, the estimation of the alignment parameters is discussed.

## 3.2 TRANSFORMATION BETWEEN REFERENCE FRAMES

Consider a particular sensor, say the $k^{th}$ sensor, where k = 1, 2. A reference frame is necessary in describing its measurements. The reference frame in which the measurements are made will be called the measurement frame. A 3D sensor measures the spherical coordinates (range, bearing, and elevation) of a target; the spherical and rectangular coordinates in the measurement frame,

and the relationship between them, are illustrated in Figure 3-1. There is also a stabilized frame associated with this sensor. The stabilized frame is aligned to the true north-south horizontal line, the true east-west horizontal line, and the axis that is orthogonal to the horizontal plane formed by the north-south and east-west lines. For a fully stabilized sensor, the measurement and stabilized frames are the same provided there are no bias errors in the gimbals; however, for unstabilized and partially stabilized sensors, both frames have the same origin, but one frame is tilted with respect to the other one.

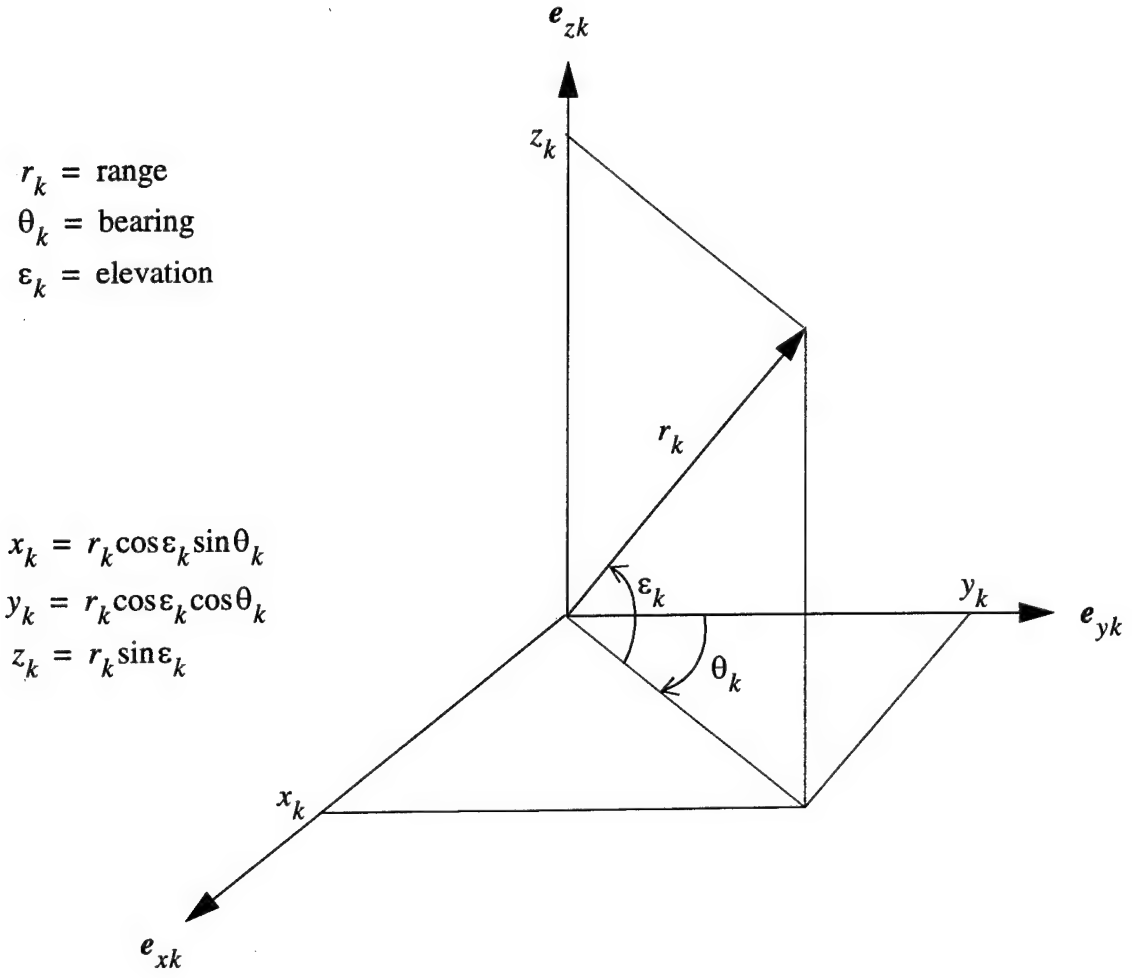


FIGURE 3-1. SPHERICAL AND RECTANGULAR COORDINATES IN K$^{TH}$ SENSOR'S MEASUREMENT FRAME

Let the column vectors $\boldsymbol{r}_k = \left[x_k\ y_k\ z_k\right]^T$ and $\boldsymbol{r}'_k = \left[x'_k\ y'_k\ z'_k\right]^T$ (the superscript $T$ denotes matrix transposition) represent the position vectors of a point (in rectangular coordinates) in the measurement and stabilized frames, respectively, of the $k^{th}$ sensor. The transformation from the measurement frame to the stabilized frame is given by

$$\boldsymbol{r}'_k = \boldsymbol{R}_k \boldsymbol{r}_k \tag{3-1}$$

where $\boldsymbol{R}_k$ is the 3x3 orthogonal matrix given by

$$\boldsymbol{R}_k = \begin{bmatrix} \cos\eta_k\cos\phi_k & \sin\psi_k\sin\eta_k\cos\phi_k - \cos\psi_k\sin\phi_k & \cos\psi_k\sin\eta_k\cos\phi_k + \sin\psi_k\sin\phi_k \\ \cos\eta_k\sin\phi_k & \sin\psi_k\sin\eta_k\sin\phi_k + \cos\psi_k\cos\phi_k & \cos\psi_k\sin\eta_k\sin\phi_k - \sin\psi_k\cos\phi_k \\ -\sin\eta_k & \cos\eta_k\sin\psi_k & \cos\eta_k\cos\psi_k \end{bmatrix} \tag{3-2}$$

and the yaw angle $\varphi_k$, the pitch angle $\eta_k$, and the roll angle $\psi_k$ are a set of Eulerian angles giving the orientation of the measurement frame in the stabilized frame (see Figure 3-2, where the primed and unprimed triads represent the stabilized and measurement frames, respectively). For a fully stabilized sensor, the yaw, pitch, and roll angles are zero provided there are no bias errors in the gimbals. For unstabilized and partially stabilized sensors, these angles need not be zero.

Let the second sensor (k = 2) be located at the point $\boldsymbol{t} = \left[t_x\ t_y\ t_z\right]^T$ in the stabilized frame of the first sensor (k = 1). Since there are no location errors, the vector $\boldsymbol{t}$ is assumed to be known. Let $\boldsymbol{r}_2$ be the position vector of a target reported by the second sensor in its measurement frame. The transformation of $\boldsymbol{r}_2$ to the measurement frame of the first sensor is given by[2]

$$\boldsymbol{r}_{12} = \boldsymbol{R}\boldsymbol{r}_2 + \boldsymbol{R}_1^T\boldsymbol{t} \tag{3-3}$$

where $\boldsymbol{r}_{12}$ represents the corresponding position vector in the measurement frame of the first sensor, and

$$\boldsymbol{R} = \boldsymbol{R}_1^T\boldsymbol{R}_2 \tag{3-4}$$

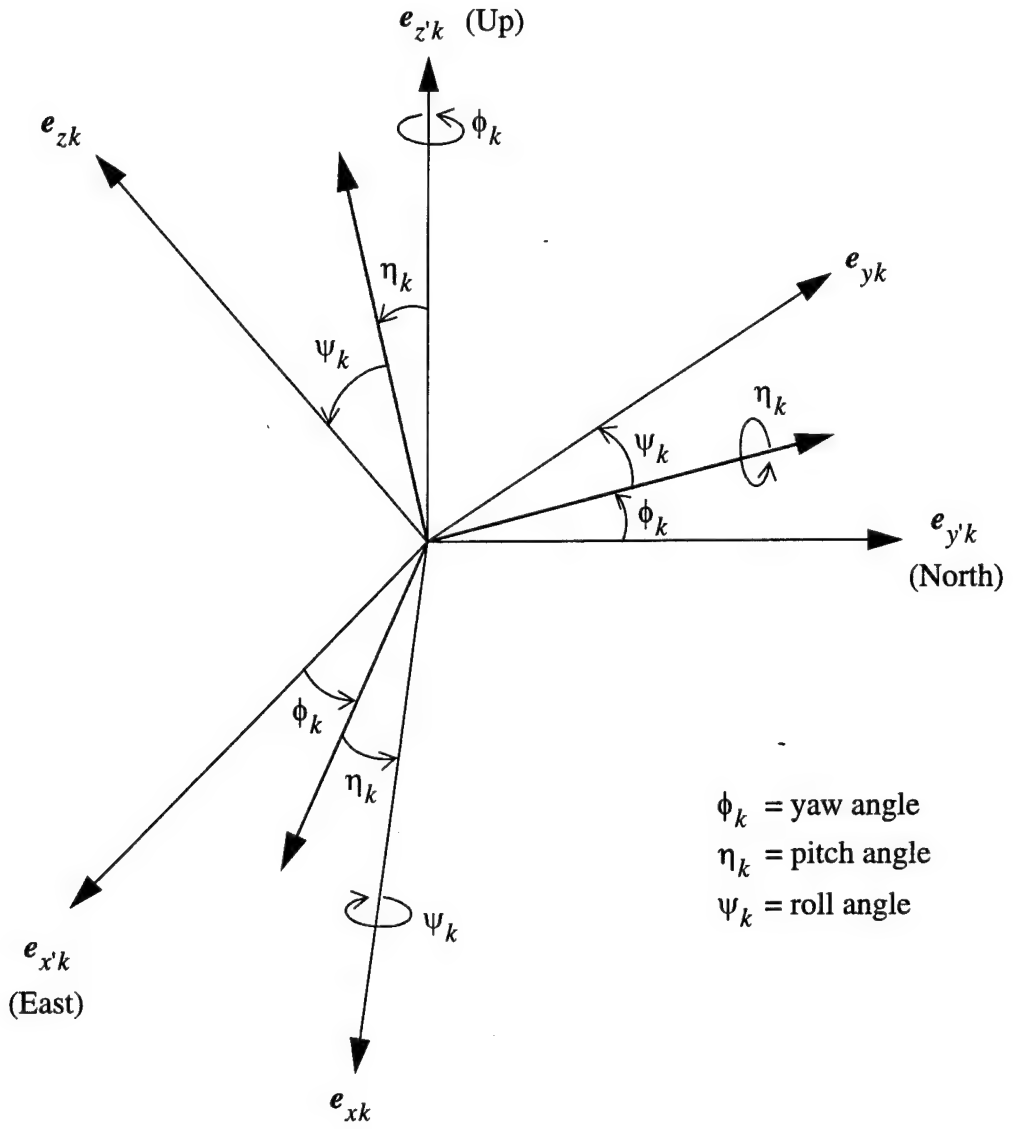


FIGURE 3-2. STABILIZED AND MEASUREMENT FRAMES FOR K[TH] SENSOR

Equation (3-3) represents the transformation of a position vector from the measurement frame of the second sensor to the measurement frame of the first sensor. The matrices $\boldsymbol{R}_1$, $\boldsymbol{R}_2$, and $\boldsymbol{R}$ depend on the values of the Eulerian angles (yaw, pitch, and roll) at the two sensors. If these Eulerian angles contain no bias errors, then, assuming there are no offset nor random errors in the sensors' measurements, $\boldsymbol{r}_{12}$ will be the same as $\boldsymbol{r}_1$, which is the position vector of the target reported by the first sensor in its measurement frame. However, if the Eulerian angles contain uncompensated bias errors, the transformation in equation (3-3) is performed using incorrect

values for the Eulerian angles and, consequently, $\boldsymbol{r}_{12}$ and $\boldsymbol{r}_1$ will be different. In this case, $\boldsymbol{r}_{12}$ does not represent the target position vector in the measurement frame of the first sensor because it was computed using incorrect values for the Eulerian angles; by definition, $\boldsymbol{r}_1$ always represents the position vector of the target in the measurement frame of the first sensor.

Although equations for transforming the velocity and acceleration vectors from the measurement frame of the second sensor to the measurement frame of the first sensor are not given in reference (2), they can be derived by taking time derivatives of equation (3-3):

$$\boldsymbol{v}_{12} = \boldsymbol{R}\boldsymbol{v}_2 + \frac{d\boldsymbol{R}}{dt}\boldsymbol{r}_2 + \frac{d\boldsymbol{R}_1^T}{dt}\boldsymbol{t} \qquad \boldsymbol{a}_{12} = \boldsymbol{R}\boldsymbol{a}_2 + 2\frac{d\boldsymbol{R}}{dt}\boldsymbol{v}_2 + \frac{d^2\boldsymbol{R}}{dt^2}\boldsymbol{r}_2 + \frac{d^2\boldsymbol{R}_1^T}{dt^2}\boldsymbol{t} \tag{3-5}$$

where $\boldsymbol{v}_2 = d\boldsymbol{r}_2/dt = \left[v_{2x}\ v_{2y}\ v_{2z}\right]^T$ and $\boldsymbol{a}_2 = d\boldsymbol{v}_2/dt = \left[a_{2x}\ a_{2y}\ a_{2z}\right]^T$ are the velocity and acceleration vectors (in rectangular coordinates) of the target reported by the second sensor in its measurement frame, $\boldsymbol{v}_{12}$ and $\boldsymbol{a}_{12}$ represent the corresponding velocity and acceleration vectors in the measurement frame of the first sensor. Similar to the discussion in the previous paragraph, if the Eulerian angles and their rates contain no bias errors, $\boldsymbol{v}_{12}$ and $\boldsymbol{a}_{12}$ will be the same as $\boldsymbol{v}_1$ and $\boldsymbol{a}_1$, respectively, which represent the velocity and acceleration vectors of the target reported by the first sensor in its measurement frame. However if the Eulerian angles or their rates contain uncompensated bias errors, the transformations in equation (3-5) are performed using incorrect values, then the velocities $\boldsymbol{v}_{12}$ and $\boldsymbol{v}_1$ will be different, and the accelerations $\boldsymbol{a}_{12}$ and $\boldsymbol{a}_1$ will also be different.

## 3.3 ALIGNMENT EQUATIONS

The attitude bias errors are modeled as additive, constant biases to the reported values of the yaw $\phi_k$, pitch $\eta_k$, and roll $\psi_k$ angles at the k$^{th}$ sensor; that is,

$$\phi_{k,\,true} = \phi_k + \Delta\phi_k \qquad \eta_{k,\,true} = \eta_k + \Delta\eta_k \qquad \psi_{k,\,true} = \psi_k + \Delta\psi_k \tag{3-6}$$

where $\phi_{k,\,true}$, $\eta_{k,\,true}$, and $\psi_{k,\,true}$ are the true values of yaw, pitch, and roll angles at the $k^{th}$ sensor, and $\Delta\phi_k$, $\Delta\eta_k$, and $\Delta\psi_k$ are the bias errors in these angles. Bias errors in the rates of the Eulerian angles are not modeled in reference (2). The $k^{th}$ sensor reports the range $r_k$, bearing $\theta_k$, and elevation $\varepsilon_k$ of a target, which are expressed in the measurement frame of the $k^{th}$ sensor. The sensor offset errors are also modeled as additive constant biases to the estimates:

$$r_{k,\,true} = r_k + \Delta r_k \qquad \theta_{k,\,true} = \theta_k + \Delta\theta_k \qquad \varepsilon_{k,\,true} = \varepsilon_k + \Delta\varepsilon_k \tag{3-7}$$

The relationships between the track data from the sensors and the alignment errors are derived assuming that the alignment errors are small quantities and using a first-order Taylor series approximation to linearize the alignment problem. The alignment is accomplished in reference (2) using five bias parameters (four angular biases and one range bias): the bearing bias $\Delta\theta$, pitch bias $\Delta\eta$, roll bias $\Delta\psi$, elevation bias $\Delta\varepsilon$, and range bias $\Delta r$. These biases are defined by

$$\begin{gathered}\Delta\theta = \Delta\phi + \Delta\theta_2 - \Delta\theta_1 \qquad \Delta\eta = \Delta\eta_1 - \Delta\eta_2 \qquad \Delta\psi = \Delta\psi_1 - \Delta\psi_2 \\ \Delta\varepsilon = \Delta\varepsilon_2 - \Delta\varepsilon_1 \qquad\qquad \Delta r = \Delta r_2 - \Delta r_1\end{gathered} \tag{3-8}$$

where

$$\Delta\phi = \Delta\phi_1 - \Delta\phi_2 \tag{3-9}$$

The definitions of these five biases imply that the alignment errors at the individual sensors cannot be determined; it is only possible to estimate the relative alignment errors between the sensors. In addition, the definition of the bearing bias $\Delta\theta$ implies that yaw bias errors cannot be distinguished from bearing offset errors.

The basic equation to align a position vector reported by the second sensor to the measurement frame of the first sensor (which is the master sensor) is given by[2]

$$\boldsymbol{r}_1 = \boldsymbol{r}_{12} + \boldsymbol{C}\boldsymbol{b} \tag{3-10}$$

where

$$\boldsymbol{b} = [\Delta\theta \;\; \Delta\eta \;\; \Delta\psi \;\; \Delta\varepsilon \;\; \Delta r]^T \tag{3-11}$$

is the bias vector and **C** is the matrix defined by

$$\boldsymbol{C}=\begin{bmatrix} y_{12} & -z_{12} & 0 & -z_{12}\sin\theta_{12} & x_{12}/r_{12} \\ -x_{12} & 0 & z_{12} & -z_{12}\cos\theta_{12} & y_{12}/r_{12} \\ 0 & x_{12} & -y_{12} & r_{12}\cos\varepsilon_{12} & z_{12}/r_{12} \end{bmatrix} \tag{3-12}$$

where $x_{12}$, $y_{12}$, and $z_{12}$ are the components of $\boldsymbol{r}_{12}$ in rectangular coordinates, and the range $r_{12}$, bearing $\theta_{12}$, and elevation $\varepsilon_{12}$ are the components of $\boldsymbol{r}_{12}$ in spherical coordinates. Assuming that the bias vector $\boldsymbol{b}$ is known, the alignment of a position vector $\boldsymbol{r}_2$ from the measurement frame of the second sensor to the measurement frame of the first sensor is accomplished by the following procedure. First, equation (3-3) is used to transform the position vector $\boldsymbol{r}_2$ reported by the second sensor using the reported values of the yaw, pitch, and roll angles; this produces $\boldsymbol{r}_{12}$. The matrix $\boldsymbol{C}$ in equation (3-12), which depends only on $\boldsymbol{r}_{12}$, is calculated. Equation (3-10) is then used to align $\boldsymbol{r}_{12}$ to the measurement frame of the first sensor. This produces $\boldsymbol{r}_1$, which is the position vector aligned to the measurement frame of the first sensor. If the first sensor reports the position vector of the target, it will coincide with $\boldsymbol{r}_1$ (in the absence of random errors). However, this alignment can be performed even if the first sensor is not tracking this particular target; of course, a common target must be used to determine $\boldsymbol{b}$.

Although equations for aligning the velocity and acceleration vectors to the measurement frame of the first sensor are not given in reference (2), they can easily be derived by taking time derivatives of equation (3-10) and using the assumption that $\boldsymbol{b}$ is a constant vector (i.e., the alignment errors are constants). The equations to align velocity and acceleration vectors reported by the second sensor to the measurement frame of the first sensor are given by

$$\boldsymbol{v}_1 = \boldsymbol{v}_{12} + \frac{d\boldsymbol{C}}{dt}\,\boldsymbol{b} \qquad\qquad \boldsymbol{a}_1 = \boldsymbol{a}_{12} + \frac{d^2\boldsymbol{C}}{dt^2}\,\boldsymbol{b} \tag{3-13}$$

Assuming that the bias vector $\boldsymbol{b}$ is known, the alignment of the velocity and acceleration vectors to the measurement frame of the first sensor is similar to the alignment of a position vector described in the previous paragraph.

## 3.4 ESTIMATION OF ALIGNMENT PARAMETERS

Since the truth is not observable, neither sensor is able to detect its own biases with respect to the truth. Thus, the bias estimation algorithm makes use of one or more tracks of targets common to both sensors to estimate *relative* correction parameters to align the data from the two sensors. This means that data from one of the sensors (sensor 2) is corrected so that it aligns in position, velocity, and acceleration with the data from the other sensor (i.e., sensor 1 is the master sensor), even though neither sensor may be aligned with the truth. To accomplish the alignment, the data from the second sensor is first expressed in the measurement frame of the master sensor using the reported values of roll, pitch and yaw, and the known translation vector between the sensors. The alignment vector $\boldsymbol{y}$ is defined as the difference between the position reported by the master sensor and the position reported by the second sensor, as expressed in the frame of the master sensor:

$$\mathbf{y} = \boldsymbol{r}_1 - \boldsymbol{r}_{12} \tag{3-14}$$

where, as defined in equation (3-10), $\boldsymbol{r}_1$ is the vector position of the target as measured by the master sensor, and $\boldsymbol{r}_{12}$ is the vector position as measured by the second sensor, but expressed in the measurement frame of the master sensor; that is, one corrects the second sensor's position vector as expressed in the master sensor's frame by adding to it the alignment vector. Using equation (3-10), the alignment vector can also be expressed as a linear combination of the sensor bias parameters,

$$\mathbf{y} = \boldsymbol{C}\boldsymbol{b}, \tag{3-15}$$

where $\boldsymbol{C}$ is defined by equation (3-12) and $\boldsymbol{b}$ is defined by equation (3-11).

Note that the bias parameters are functions of the differences between the biases, not the biases themselves, because the individual biases are not observable. Also, there is coupling between some of the bias equations, which results in the bias parameters being functions of more than one of the six biases. For example, as seen in equation (3-8), the bearing bias and the yaw bias are inseparable, because they both result from offsets in the same plane. Thus, the parameter $\Delta\theta$ is a function of both the bearing and yaw biases.

Assume that both sensors are tracking a common target, and let $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ be the position vectors of this target as reported by the first and second sensors, respectively, at the same point in time. Then $\boldsymbol{r}_{12}$ and $\boldsymbol{C}$ can be computed, and equation (3-10) can be used as a measurement equation of the bias vector by expressing it as

$$\boldsymbol{y} = \boldsymbol{r}_1 - \boldsymbol{r}_{12} = \boldsymbol{C}\boldsymbol{b} + \boldsymbol{e} \tag{3-16}$$

where the random measurement error vector $\boldsymbol{e}$ is now included in the problem. It is possible to apply Kalman filtering techniques to estimate the bias parameter vector $\boldsymbol{b}$ recursively. Since the errors in the sensors' measurements are assumed to be white noise sequences, the error $\boldsymbol{e}$ is also a white noise sequence when position measurements are used to obtain $\boldsymbol{y}$. However, when $\boldsymbol{y}$ is obtained using position estimates from the tracking filters associated with the sensors, $\boldsymbol{e}$ is not a white noise sequence. This is due to the time-correlated estimation errors from the tracking filters. In reference (5), it was assumed, for simplicity, that $\boldsymbol{e}$ is a white noise sequence so that standard estimation results could be easily applied to the problem.

Note that $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and therefore $\boldsymbol{r}_{12}$ are assumed to be time-coincident quantities. Since the sensors are asynchronous, the state estimate reported by the second sensor must be time aligned to the estimate reported by the master sensor. This is done by predicting the most recent estimate from the second sensor to the time of the current estimate for the master sensor. To improve the accuracy of the predicted estimate, one-step, fixed-lag smoothing may also be performed. The effectiveness of these time-alignment techniques is studied in the next section in a series of simulations.

It is possible to separate the range bias equation from the angular bias equation, and to estimate them using separate filters. This is advantageous because the contribution of the range bias $\Delta r$ to the alignment error is usually much smaller than that due to the angular biases. By decoupling, difficulties in estimating the range bias are avoided. It can be shown that the measurement equation in equation (3-16) can be reduced to the following two *decoupled* measurement equations:[2]

$$m_r = \Delta r + e_r \qquad\qquad \boldsymbol{z} = \boldsymbol{H}\boldsymbol{d} + \boldsymbol{v} \tag{3-17}$$

where the measurement $m_r$ is defined by $m_r = r_1 - r_{12}$, $e_r$ is the measurement error in $m_r$, the measurement $z$ is defined by

$$z = \left[(\theta_1 - \theta_{12}) \quad (\varepsilon_1 - \varepsilon_{12})\right]^T, \tag{3-18}$$

$v$ is the measurement error in $z$, $d$ is the angular bias vector defined by

$$d = \left[\Delta\theta \ \Delta\eta \ \Delta\psi \ \Delta\varepsilon\right]^T, \tag{3-19}$$

and $H$ is the matrix given by

$$H = \begin{bmatrix} 1 & -\tan\varepsilon_{12}\cos\theta_{12} & -\tan\varepsilon_{12}\sin\theta_{12} & 0 \\ 0 & \sin\theta_{12} & -\cos\theta_{12} & 1 \end{bmatrix} \tag{3-20}$$

Clearly, $m_r = \Delta r + e_r$ is the measurement equation for the range bias, and $z = Hd + v$ is the measurement equation for the angular biases.

The variance $\sigma_r^2$ of the measurement error $e_r$ is approximated by $\sigma_r^2 \approx \sigma_{r1}^2 + \sigma_{r2}^2$, where $\sigma_{r1}^2$ and $\sigma_{r2}^2$ are the variances in the ranges that are computed using the error covariance matrices from the tracking filters for the first and second sensors, respectively. Note that the computation of $\sigma_r^2$ ignores the correlation between the estimation errors in the tracks of the common target from the two sensors that arises from the common process noise in the dynamics;[7,8] that is, the tracking errors from the two sensors for a common target are dependent, but they will be assumed independent for simplicity. Similarly, using this assumption of independence, the covariance matrix $W$ of the measurement error $v$ is approximated by

$$W \approx \begin{bmatrix} (\sigma_{\theta 1}^2 + \sigma_{\theta 2}^2) & (\rho_1\sigma_{\theta 1}\sigma_{\varepsilon 1} + \rho_2\sigma_{\theta 2}\sigma_{\varepsilon 2}) \\ (\rho_1\sigma_{\theta 1}\sigma_{\varepsilon 1} + \rho_2\sigma_{\theta 2}\sigma_{\varepsilon 2}) & (\sigma_{\varepsilon 1}^2 + \sigma_{\varepsilon 2}^2) \end{bmatrix} \tag{3-21}$$

where $\sigma_{\theta k}^2$, $\sigma_{\varepsilon k}^2$, and $\rho_k$ are the azimuth variance, elevation variance, and correlation coefficient between the azimuth and elevation that are computed using the error covariance matrix from the tracking filter for the $k^{th}$ sensor.

# 4.0 METHOD OF SIMULATION

## 4.1 INTRODUCTION

The simulation of the targets, tracking, and alignment process are described in this section. Also, the implementation of the bias parameter estimation algorithm is discussed and further developed. In the simulation, the bias filters run concurrently with the tracking filters. After initializing the tracking filters with several data points, a few more data points are required to initialize the concurrently running bias estimation filters. The bias estimation algorithm requires state estimates or measurements from each of the two sensors, and for one or more common targets. Using more than one common target provides quicker convergence because the system of equations to be solved is then specified by more equations in the same number of unknowns. Once the bias estimation has been initialized, the alignment of the second sensor track can proceed along with the tracking, so that at each time step, a corrected state estimate is produced. The corrected state estimate from the second sensor and the state estimate from the master sensor can then be passed to an association algorithm. These processes are discussed in more detail below.

## 4.2 FORMATION OF SIMULATED TRAJECTORIES

To simulate the alignment of two sensors, the first requirement is to obtain simulated *true* trajectories for two or three targets. One or two of the targets will be used to estimate the bias parameters, and the third one will be corrected using the estimated alignment vector. In theory, once it has been estimated correctly, the alignment should work for the track of any target being observed by the second sensor, not just for the one(s) used to estimate the biases. The simulated trajectories are initially generated in Cartesian coordinates. The true trajectories are then altered so that they realistically represent noisy, biased tracks from separate sensors, in the following

manner. First, the data for the second sensor is translated in position by subtracting from it the constant vector $\boldsymbol{t}$, representing the location of sensor 2 relative to sensor 1. The measurements of the master sensor are then computed at a shifted set of times to simulate asynchronous measurements. Each sensors' data is transformed into its own measurement frame by a rotation corresponding to the magnitudes of the specified roll, pitch and yaw biases. The trajectories are then sampled at a desired rate, assuming the sample rate of the original trajectory is higher than the one desired. The preliminary processing of each trajectory is completed by converting the Cartesian measurements to spherical coordinates, adding to each spherical coordinate zero-mean Gaussian noise with a standard deviation corresponding to the assumed accuracy of the radar, and adding specified constant biases to the range, bearing and elevation from each sensor. The data for the three trajectories are then ready to be passed into the tracking loop, to simulate the processes of tracking, time alignment, bias estimation, and track association.

## 4.3 TRACKING

Each target is tracked by each sensor using either a single model Kalman filter or an Interactive Multiple Model (IMM) algorithm.[6] The filters use the current spherical coordinate sensor measurements, and the previous Cartesian state estimates and covariances to produce updated Cartesian state estimates and covariances. The single model filter can employ any practical dynamical model, such as CV or Constant Acceleration (CA). The IMM algorithm in this report uses both of these models, producing model-conditioned state estimates and covariances as well as a combined state estimate and covariance.

## 4.4 TIME ALIGNMENT

In this section, two techniques used in the time alignment of data from two asynchronous sensors are discussed. The first is *prediction* of a state estimate to a future time, and the second, which may be applied to the predicted estimate to improve its accuracy, is *one-step, fixed-lag smoothing*.

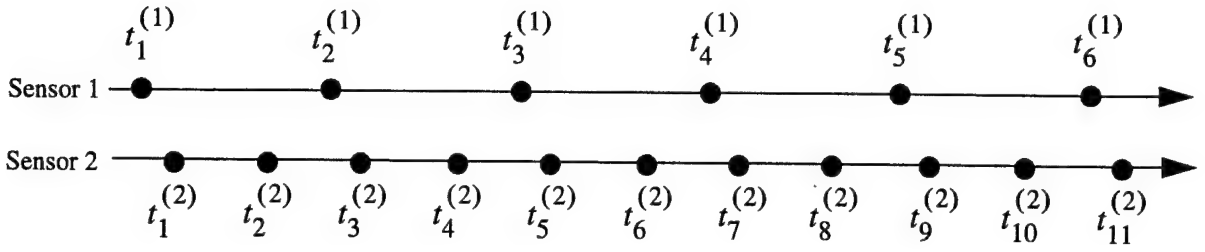


FIGURE 4-1. TIMES OF STATE ESTIMATES REPORTED BY TWO SENSORS FOR A SINGLE COMMON TARGET

The principal difficulty in alignment is time-translating the data from the two sensors to a common point of time. This situation is illustrated in Figure 4-1, where the two sensors are providing state estimates for a single target at different times and at different rates. In Figure 4-1, $t_j^{(i)}$ denotes the time that the $j^{th}$ state estimate for the target was reported by sensor $i$. One method to handle this problem is to predict the state estimates from the second sensor forward in time to match the times of the first sensor. For example, the state at $t_2^{(2)}$ from the second sensor would be predicted to $t_2^{(1)}$, the state at $t_4^{(2)}$ from the second sensor would be predicted to $t_3^{(1)}$, etc. A second method would use both the one-step predictor and a one-step, fixed-lag smoother. For example, the state at $t_2^{(2)}$ from the second sensor would be predicted to $t_2^{(1)}$, then the measurement at $t_3^{(2)}$ from the second sensor would be used to improve this predicted estimate at $t_2^{(1)}$ (i.e., one-step smoothing); the state at $t_4^{(2)}$ from the second sensor would be predicted to $t_3^{(1)}$, then the measurement at $t_5^{(2)}$ from the second sensor would be used to improve this predicted estimate at $t_3^{(1)}$; etc. Since one-step, fixed-lag smoothing provides an estimate that lags the most recent measurement by one sampling period, it is not a real-time process. However, it is near real-time and the time lag will not be a serious problem in estimating the alignment errors because the alignment errors are assumed constant or slowly varying with time. Also, smoothing

will provide additional noise variance reduction over prediction and filtering, which is important because the alignment algorithm uses differences in target positions as reported by the sensors to estimate the alignment error

## 4.5 PREDICTION

The bias estimation algorithm does not permit the use of asynchronous estimates or measurements from the two sensors. Therefore, after computing the filtered state estimates for both sensors, it is necessary to time align them; i.e., use the current state estimate, covariance, and dynamical model for one of the sensors to estimate the state at a short time in the future at which an estimate for the other sensor already exists. An algorithm that performs this task is called a *predictor.* Both single and multiple-model predictors are used in the simulation, depending on which tracking filter is used. The single model predictor takes the current Cartesian state vector to be predicted, its covariance matrix, the time at which the predicted state is to be computed, and the process noise variance, and produces the predicted Cartesian state vector and the predicted covariance matrix at the predicted time. The multiple-model predictor uses the current model-conditioned Cartesian states and covariances from the IMM algorithm and produces the predicted model-conditioned Cartesian state estimates and covariances and a combined predicted Cartesian state estimate and covariance for output.

### 4.5.1 Single-Model Prediction

For the single model case, the linear system dynamics are modeled as

$$X_{k+1} = F_k X_k + G_k W_k \tag{4-1}$$

with observations

$$Z_k = H_k X_k + V_k \tag{4-2}$$

where $X_k$ is the system state at time $t_k$, $Z_k$ is the measurement vector, and $F_k$, $G_k$, and $H_k$ are the system matrices. The $W_k$ and $V_k$ are mutually uncorrelated, zero-mean, white Gaussian

errors with covariance matrices $Q_k$ and $R_k$, respectively. Single-model prediction is governed by the following equations

State estimate prediction: $$X_{k+1|k} = F_k X_{k|k} \tag{4-3}$$

Covariance prediction: $$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k Q_k G_k^T \tag{4-4}$$

where $X_{k|k}$ and $P_{k|k}$ are the filtered state estimate and associated error covariance, respectively, and $X_{k+1|k}$ and $P_{k+1|k}$ are the one-step predicted state estimate and associated error covariance, respectively.

### 4.5.2 Multiple-Model, One-Step Prediction Algorithm for Alignment

In the IMM algorithm,[6] the system (called a Markovian switching system) is assumed to obey one of a finite number of models and there is a given probability of switching between the models. The event that model $j$ ($j = 1, \ldots, n$) is in effect during the sampling period ending at time $t_k$ will be denoted by $M_k^j$. The system dynamics and observations are modeled as

$$X_k = F_{k-1}^j X_{k-1} + G_{k-1}^j W_{k-1}^j \tag{4-5}$$

and

$$Z_k = H_k^j X_k + V_k^j \tag{4-6}$$

where $X_k$ is the system state at time $t_k$, $Z_k$ is the measurement vector, and $F_{k-1}^j$, $G_{k-1}^j$, and $H_k^j$ are the system matrices for $M_k^j$. The $W_{k-1}^j$ and $V_k^j$ are mutually uncorrelated zero-mean white Gaussian errors with covariance matrices $Q_{k-1}^j$ and $R_k^j$, respectively. The switching between the models is assumed to be a Markov process with known transition probabilities $p_{ij} = P\{M_k^j | M_{k-1}^i\}$.

In the one-step predictor, the estimate of the state $X_k$ at time $t_k$ is computed using the set of past data $Z_1^{k-1} = \{Z_1, \ldots, Z_{k-2}, Z_{k-1}\}$, and this estimate is denoted by $X_{k|k-1}$. For Markovian switching systems, the one-step predicted estimate can be approximated by computing it under each possible model hypothesis over the current sampling period. Using the total probability theorem for the $n$ model hypotheses in effect over the current sampling period, the density function of the state $X_k$ given $Z_1^{k-1}$ can be expressed as

$$f\left[X_k \middle| Z_1^{k-1}\right] = \sum_{j=1}^{n} \bar{c}_j \, f\left[X_k \middle| M_k^j, Z_1^{k-1}\right] \tag{4-7}$$

where $\bar{c}_j = P\{M_k^j \middle| Z_1^{k-1}\}$ is the one-step predicted probability for $M_k^j$, and $f\left[X_k \middle| M_k^j, Z_1^{k-1}\right]$ is the model-conditioned density function of $X_k$ matched to $M_k^j$. Both $\bar{c}_j$ and the model-conditioned estimate $X_{k|k-1}^j$ based on $f\left[X_k \middle| M_k^j, Z_1^{k-1}\right]$ are computed in the IMM filtering algorithm. The only calculation needed in the one-step predictor that is not performed in the IMM filtering algorithm is the computation of the one-step predicted estimate $X_{k|k-1}$ based on $f\left[X_k \middle| Z_1^{k-1}\right]$ when all models are considered. Equation (4-7) represents $f\left[X_k \middle| Z_1^{k-1}\right]$ as a mixture of densities, which are assumed Gaussian. This Gaussian mixture can be approximated by a single Gaussian,[6] which gives $X_{k|k-1}$ as a probabilistic sum of the model-conditioned estimates $X_{k|k-1}^j$ and is presented as follows in *Step* 3.

One cycle of the multiple-model, one-step prediction algorithm for $n$ models is summarized in the following three steps.

*Step 1: Mixing of the State Estimates*

The prediction process starts with the filtered state estimates $X_{k-1|k-1}^i$ matched to $M_{k-1}^i$, associated error covariances $P_{k-1|k-1}^i$, and model probabilities $\mu_{k-1}^i$ from the IMM filter. The

mixed state estimate $X^{0j}_{k-1|k-1}$ for $M^j_k$ is computed as

$$X^{0j}_{k-1|k-1} = \sum_{i=1}^{n} X^{i}_{k-1|k-1} \mu^{i|j}_{k-1|k-1} \tag{4-8}$$

where

$$\mu^{i|j}_{k-1|k-1} = \frac{1}{\bar{c}_j} p_{ij} \mu^{i}_{k-1} \tag{4-9}$$

and the normalization constant $\bar{c}_j$ is given by

$$\bar{c}_j = \sum_{i=1}^{n} p_{ij} \mu^{i}_{k-1} \tag{4-10}$$

where $p_{ij}$ is the assumed transition probability for switching from model $i$ to model $j$. The error covariance $P^{0j}_{k-1|k-1}$ associated with $X^{0j}_{k-1|k-1}$ is computed as

$$P^{0j}_{k-1|k-1} = \sum_{i=1}^{n} \mu^{i|j}_{k-1|k-1} \left[ P^{i}_{k-1|k-1} + \left[ X^{i}_{k-1|k-1} - X^{0j}_{k-1|k-1} \right] \left[ X^{i}_{k-1|k-1} - X^{0j}_{k-1|k-1} \right]^T \right] \tag{4-11}$$

*Step 2: Model-Conditioned Predictions*

The one-step predicted estimate $X^j_{k|k-1}$ matched to model $M^j_k$, and associated covariance $P^j_{k|k-1}$, are computed as

$$X^j_{k|k-1} = F^j_{k-1} X^{0j}_{k-1|k-1} \tag{4-12}$$

$$P^j_{k|k-1} = F^j_{k-1} P^{0j}_{k-1|k-1} (F^j_{k-1})^T + G^j_{k-1} Q^j_{k-1} (G^j_{k-1})^T \tag{4-13}$$

The predicted model probability for $M^j_k$ is given by $\bar{c}_j$, which is calculated in Step 1.

*Step 3: Combination of State Estimates*

The one-step predicted state estimate $X_{k|k-1}$ and error covariance $P_{k|k-1}$ for output are given by

$$X_{k|k-1} = \sum_{j=1}^{n} \bar{c}_j X^j_{k|k-1} \tag{4-14}$$

$$P_{k|k-1} = \sum_{j=1}^{n} \bar{c}_j \left[ P^j_{k|k-1} + \left[ X^j_{k|k-1} - X_{k|k-1} \right] \left[ X^j_{k|k-1} - X_{k|k-1} \right]^T \right] \tag{4-15}$$

The time translation of the estimates from the second sensor using the multiple-model, one-step predictor is implemented as follows. The time of the most current filtered estimate from the first sensor (i.e., the master sensor) is obtained; denote this estimate by $X^{(1)}_{k|k}$, which is valid at time $t^{(1)}_k$, the time of the first sensor's most current estimate. The model-conditioned filtered estimates from the second sensor that immediately precedes $t^{(1)}_k$ are obtained; denote these model-conditioned estimates by $X^{(2),j}_{l|l}$, which are valid at time $t^{(2)}_l$, the time of the second sensors' estimates that immediately precede $t^{(1)}_k$. The multiple-model, one-step prediction algorithm is used to predict the second sensors' estimates from time $t^{(2)}_l$ to $t^{(1)}_k$ (by letting $t_{k-1} = t^{(2)}_l$, $t_k = t^{(1)}_k$, and $X^j_{k-1|k-1} = X^{(2),j}_{l|l}$ in the algorithm), and the model-conditioned predicted estimates $X^{(2),j}_{k|k-1}$ and the output predicted estimate $X^{(2)}_{k|k-1}$ for the second sensor at time $t^{(1)}_k$ are obtained. The time-coincident estimates $X^{(1)}_{k|k}$ and $X^{(2)}_{k|k-1}$ from the two sensors are then passed to the alignment algorithm described previously.

## 4.6 ONE-STEP, FIXED-LAG SMOOTHING

One-step, fixed-lag smoothing uses the past and present data to compute the state estimate one-step behind the present time. Since one-step, fixed-lag smoothing estimates a past state, it is not a real-time process; however it is near real time since it provides an estimate that lags the

current measurement by one sampling period. To improve the accuracy of the predicted estimates, this type of smoothing is employed in the simulation. As with the predictor, there are both single and multiple model smoothers, depending upon the tracking filter that is in use. If smoothing is used, the predicted estimates are smoothed prior to the estimation of the bias parameters. Fixed-lag smoothers are discussed in detail in reference (6), and a detailed description of one-step, fixed-lag smoothers for Markovian switching systems is given in reference (4).

The single model one-step, fixed-lag smoothing algorithm, as used in the simulation, is slightly different than the standard one-step, fixed-lag smoothing algorithm. In the standard smoothing algorithm, the smoothed estimate is computed as a correction to a filtered estimate. However, in the smoothing algorithm used in the simulation, the smoothed estimate is computed as a correction to a predicted estimate. The reason is that the one-step predictor is used to time translate the estimate from the second sensor; then, the one-step smoother is used to improve the accuracy of this predicted estimate by processing the *future* measurement from one-step ahead. The one-step, fixed-lag smoothing algorithm used in the simulation is briefly summarized.

### 4.6.1 Single-Model, One-Step, Fixed-Lag Smoothing Algorithm

*Step 1*: *Predict the State and Covariance*

Start with the one-step predicted estimate $X_{k|k-1}$ and associated error covariance $P_{k|k-1}$ from the one-step predictor at time $t_k$. These quantities are then predicted to time $t_{k+1}$ using

$$X_{k+1|k-1} = F_k X_{k|k-1} \tag{4-16}$$

$$P_{k+1|k-1} = F_k P_{k|k-1} F_k^T + G_k Q_k G_k^T \tag{4-17}$$

*Step 2*: *Compute the Residual*

The residual $\tilde{Z}_{k+1|k-1}$ and its covariance $S_{k+1|k-1}$ are computed by

$$\tilde{Z}_{k+1|k-1} = Z_{k+1} - H_{k+1} X_{k+1|k-1} \tag{4-18}$$

$$S_{k+1|k-1} = H_{k+1} P_{k+1|k-1} H_{k+1}^T + R_{k+1} \tag{4-19}$$

*Step 3*: *Compute the Smoothing Gain* $W_{k+1|k-1}$

$$W_{k+1|k-1} = P_{k|k-1} F_k^T H_{k+1}^T S_{k+1|k-1}^{-1} \tag{4-20}$$

*Step 4: Update the Smoothed State and Smoothed Covariance*

The smoothed state $X_{k|k+1}$ and associated error covariance $P_{k|k+1}$ are computed by

$$X_{k|k+1} = X_{k|k-1} + W_{k+1|k-1} \tilde{Z}_{k+1|k-1} \tag{4-21}$$

$$P_{k|k+1} = P_{k|k-1} - W_{k+1|k-1} S_{k+1|k-1} W_{k+1|k-1}^T \tag{4-22}$$

The multiple-model smoothing algorithm is more complicated because it requires model-conditioned updates for the state estimate and covariance matched to each model, model probability updates, and combination of the state estimates. The multiple-model smoother also produces a combined state estimate and covariance for output. The following is a summary of the multiple-model, one-step, fixed-lag smoothing algorithm.

### 4.6.2 Multiple-Model One-Step Fixed-Lag Smoothing Algorithm

Derivation and detailed explanation of the one-step fixed-lag smoothing algorithm for Markovian switching systems are given in reference (4). The only difference between the smoothing algorithm in reference (4) and the one used in this report is that the smoothed estimate in reference (4) is computed as a correction to a filtered estimate, while we will take the smoothed estimate as a correction to a predicted estimate. The reason for this is that the multiple-model, one-step predictor will be used to time translate the estimates from the second sensor as in the previous section; then, the one-step smoother will be used to improve the accuracy of the predicted estimate by processing the *future* measurement from the stage one-step ahead. One cycle of the one-step, fixed-lag smoothing algorithm for $n$ models is summarized in the following four steps.

*Step* 1: *Mixing of the Residuals*

Start with the one-step predicted estimates $X_{k|k-1}^j$ from the multiple-model, one-step predictor matched to model $M_k^j$ at time $t_k$, the associated error covariances $P_{k|k-1}^j$, and the

corresponding model probabilities $\bar{c}_j$. Then, for each model $M_{k+1}^i$ in effect over the sampling period ending at time $t_{k+1}$, compute

$$X_{k+1|k-1}^{i|j} = F_k^i X_{k|k-1}^j \tag{4-23}$$

$$P_{k+1|k-1}^{i|j} = F_k^i P_{k|k-1}^j (F_k^i)^T + G_k^i Q_k^i (G_k^i)^T \tag{4-24}$$

$$\tilde{Z}_{k+1|k-1}^{i|j} = Z_{k+1} - H_{k+1}^i X_{k+1|k-1}^{i|j} \tag{4-25}$$

$$S_{k+1|k-1}^{i|j} = H_{k+1}^i P_{k+1|k-1}^{i|j} (H_{k+1}^i)^T + R_{k+1}^i \tag{4-26}$$

$$\Lambda_{k+1|k-1}^{i|j} = \frac{1}{\sqrt{det\left(2\pi S_{k+1|k-1}^{i|j}\right)}} exp\left[-\frac{1}{2}\left(\tilde{Z}_{k+1|k-1}^{i|j}\right)^T \left(S_{k+1|k-1}^{i|j}\right)^{-1} \tilde{Z}_{k+1|k-1}^{i|j}\right] \tag{4-27}$$

The mixed values input to the smoother matched to model $M_k^j$ are computed as

$$\tilde{Z}_{k+1|k-1}^j = \sum_{i=1}^n p_{ji} \tilde{Z}_{k+1|k-1}^{i|j} \tag{4-28}$$

$$S_{k+1|k-1}^j = \sum_{i=1}^n p_{ji}\left[S_{k+1|k-1}^{i|j} + \left[\tilde{Z}_{k+1|k-1}^{i|j} - \tilde{Z}_{k+1|k-1}^j\right]\left[\tilde{Z}_{k+1|k-1}^{i|j} - \tilde{Z}_{k+1|k-1}^j\right]^T\right] \tag{4-29}$$

$$\Lambda_{k+1|k-1}^j = \sum_{i=1}^n p_{ji} \Lambda_{k+1|k-1}^{i|j} \tag{4-30}$$

$$T_{k+1}^j = \sum_{i=1}^n p_{ji} (F_k^i)^T (H_{k+1}^i)^T \tag{4-31}$$

where $p_{ji}$ is the assumed transition probability for switching from model $M_k^j$ to model $M_{k+1}^i$.

*Step* 2: *Model-Conditioned Updates*

The one-step, fixed-lag smoothed estimate $X_{k|k+1}^j$ and covariance $P_{k|k+1}^j$ matched to model $M_k^j$ are computed as

$$X^j_{k|k+1} = X^j_{k|k-1} + W^j_{k+1|k-1}\tilde{Z}^j_{k+1|k-1} \tag{4-32}$$

$$P^j_{k|k+1} = P^j_{k|k-1} - W^j_{k+1|k-1}S^j_{k+1|k-1}\left(W^j_{k+1|k-1}\right)^T \tag{4-33}$$

where the smoothing gain is given by

$$W^j_{k+1|k-1} = P^j_{k|k-1}T^j_{k+1}\left(S^j_{k+1|k-1}\right)^{-1} \tag{4-34}$$

*Step* 3: *Model-Probability Updates*

The one-step, fixed-lag smoothed model probability $\mu^j_{k|k+1}$ matched to model $M^j_k$ is computed as

$$\mu^j_{k|k+1} = \frac{1}{d}\bar{c}_j\Lambda^j_{k+1|k-1} \tag{4-35}$$

where

$$d = \sum_{j=1}^{n}\bar{c}_j\Lambda^j_{k+1|k-1} \tag{4-36}$$

and $\bar{c}_j$ is the one-step predicted model probability for model $M^j_k$.

*Step* 4: *Combination of State Estimates*

The output one-step, fixed-lag smoothed estimate $X_{k|k+1}$ and corresponding error covariance $P_{k|k+1}$ are computed as

$$X_{k|k+1} = \sum_{j=1}^{n}\mu^j_{k|k+1}X^j_{k|k+1} \tag{4-37}$$

$$P_{k|k+1} = \sum_{j=1}^{n}\mu^j_{k|k+1}\left[P^j_{k|k+1} + \left[X^j_{k|k+1} - X_{k|k+1}\right]\left[X^j_{k|k+1} - X_{k|k+1}\right]^T\right] \tag{4-38}$$

The multiple-model, one-step smoother is implemented as follows. The time of the most current filtered estimate from the first sensor (i.e., the master sensor) is obtained; denote this estimate by $X^{(1)}_{k|k}$, which is valid at time $t^{(1)}_k$, the time of the first sensor's most current estimate.

Identical to the discussion in the previous section, the multiple-model, one-step predictor is used to obtain model-conditioned, one-step predicted estimates $X_{k|k-1}^{(2),j}$ for the second sensor, which are valid at time $t_k^{(1)}$. The measurement from the second sensor that occurs immediately after $t_k^{(1)}$ is obtained; denote this measurement by $Z_{k+1}^{(2)}$, which occurs at $t_{k+1}^{(2)}$. The multiple-model, one-step, fixed-lag smoothing algorithm above is used to process this measurement, and the output smoothed estimate $X_{k|k+1}^{(2)}$ for the second sensor at time $t_k^{(1)}$ is obtained. The time-coincident estimates $X_{k|k}^{(1)}$ and $X_{k|k+1}^{(2)}$ from the two sensors are then passed to the alignment algorithm.

Although both the single-model and the multiple-model algorithms assumed a linear measurement model, it is straightforward to include a nonlinear measurement model in these algorithms. In the simulations in this report, the tracking algorithms actually used a nonlinear measurement model, which represented the transformation from Cartesian to spherical coordinates.

## 4.7 BIAS PARAMETER ESTIMATION

The bias estimation algorithm requires time-coincident state estimates, or synchronous raw measurements from each of the sensors (or a measurement from one sensor and a time-aligned estimate from the other). For asynchronous estimates, the prediction and smoothing algorithms can be used to provide time-coincident estimates to the alignment algorithm. Since the estimation is performed iteratively, the bias parameter estimates and covariances from the previous step are also required, along with the assumed standard deviations of the measurements, the process noise, and the vector $\boldsymbol{t}$ that gives the relative location between the sensors. The algorithm produces updated bias parameters and covariances. The model for the biases is given by

Dynamics: $$\boldsymbol{b}_k = \boldsymbol{b}_{k-1} + \boldsymbol{u}_{k-1} \tag{4-39}$$

Measurements: $$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{b} + \boldsymbol{e} \tag{4-40}$$

where $\boldsymbol{y} = \boldsymbol{r}_1 - \boldsymbol{r}_{12}$ is the measurement of $\boldsymbol{b}$ and $\boldsymbol{e}$ is the random error in the measurement. Note that $\boldsymbol{r}_1$ and $\boldsymbol{r}_{12}$ are assumed to be time-coincident quantities.

It was found that decoupling the range and angular biases during estimation produced more accurate results. This was done in reference (2) by algebraically manipulating the vector bias equation in equation (4-40), and making small angle approximations. The bias parameter estimation algorithm proceeds as follows:

*Step* 1. *Express the Position Vector of the Target Reported by Sensor 2 in the Measurement Frame of the Master Sensor (Sensor 1).*

Expressing the position vector of the target reported by sensor 2 in the measurement frame of the master sensor requires that the sensor 2 position vector be transformed into its own stabilized frame, then translated to the stabilized frame of the master sensor, and finally rotated into the measurement frame of the master sensor:

$$\boldsymbol{r}_{12} = \boldsymbol{R}_1^T \boldsymbol{R}_2 \boldsymbol{r}_2 + \boldsymbol{R}_1^T \boldsymbol{t}. \tag{4-41}$$

This equation was simplified in reference (2) by expanding $\boldsymbol{R}_i$ in a Taylor series and using the first-order approximation

$$\boldsymbol{R}_i = \boldsymbol{I} + d\boldsymbol{R}_i, \tag{4-42}$$

where $\boldsymbol{I}$ is the identity matrix and $d\boldsymbol{R}_i$ is the differential matrix

$$d\boldsymbol{R}_i = \begin{bmatrix} 0 & -\varphi_i & \eta_i \\ \varphi_i & 0 & -\psi_i \\ -\eta_i & \psi_i & 0 \end{bmatrix} \tag{4-43}$$

where $\varphi_i$, $\eta_i$, and $\psi_i$ are the roll, pitch, and yaw, respectively, reported for sensor $i$. By letting $\boldsymbol{R} = \boldsymbol{R}_1^T \boldsymbol{R}_2 \approx \boldsymbol{I} + d\boldsymbol{R}$, the differential matrix $d\boldsymbol{R}$ is obtained

$$d\boldsymbol{R} = d\boldsymbol{R}_1^T + d\boldsymbol{R}_2 = \begin{bmatrix} 0 & \varphi_1 - \varphi_2 & -(\eta_1 - \eta_2) \\ -(\varphi_1 - \varphi_2) & 0 & \psi_1 - \psi_2 \\ \eta_1 - \eta_2 & -(\psi_1 - \psi_2) & 0 \end{bmatrix} \tag{4-44}$$

This permits a first-order approximation to the transformation from the measurement frame of sensor 2 to the measurement frame of the master sensor (sensor 1)

$$\boldsymbol{r}_{12} \approx \boldsymbol{r}_2 + \boldsymbol{t} + d\boldsymbol{R}\boldsymbol{r}_2 + d\boldsymbol{R}_1^T\boldsymbol{t} \tag{4-45}$$

For simplicity, and with no loss of generality, it is assumed in the simulation that the reported values of roll, pitch and yaw for each sensor are zero; i.e., the reported measurement frame and the stabilized frames coincide for each sensor.

*Step 2. Compute the Measured Angular Bias Vector and the Measured Range Bias*

Since the position vectors and the translation vector are passed to the bias estimation algorithm in Cartesian coordinates, it is necessary to transform them into spherical coordinates. It is noted that the angular biases have been decoupled from the range bias so that they can be computed separately.[2] The bias measurements are given by

$$Z_{\text{angular}} = \begin{bmatrix} \theta_1 - \theta_{12} \\ \varepsilon_1 - \varepsilon_{12} \end{bmatrix} \qquad Z_{\text{range}} = r_1 - r_{12} \tag{4-46}$$

Here, $\theta_1$, $\varepsilon_1$, and $r_1$ are the azimuth, elevation and range, respectively, of the master sensor, and $\theta_{12}$, $\varepsilon_{12}$, and $r_{12}$, are the azimuth, elevation and range, respectively, of the second sensor, but expressed in the measurement frame of the master sensor. Note that this is the single target formulation. If two targets are used, the measurement vectors and other related matrix quantities are *stacked*, and computations proceed in the same manner.

*Step 3. Perform the Kalman Filtering Steps to Estimate the Angular Bias Parameters*

At each step, compute the following:

$$P_{k+1|k} = P_{k|k} + Q \tag{4-47}$$

$$S_{k+1} = L_{k+1} P_{k+1|k} L_{k+1}^T + W \tag{4-48}$$

$$K_{k+1} = P_{k+1|k} L_{k+1}^T S_{k+1}^{-1} \tag{4-49}$$

$$P_{k+1|k+1} = (I - K_{k+1} L_{k+1}) P_{k+1|k} \tag{4-50}$$

$$\tilde{Z}_{\text{angular}} = Z_{\text{angular}} - C_{k+1}\boldsymbol{b}_{k+1|k} \tag{4-51}$$

$$\boldsymbol{b}_{k+1|k+1} = \boldsymbol{b}_{k+1|k} + K_{k+1}\tilde{Z}_{\text{angular}} \tag{4-52}$$

where $P$ is the covariance of the angular bias state vector, $Q$ is the process noise covariance, $S$ is the innovation covariance, $K$ is the Kalman gain, $W$ is the covariance of the random error in the

measurement of the angular bias vector (see equation (3-21)), $L$ is the observation matrix given by

$$L = \begin{bmatrix} 1 & -\tan(\varepsilon_{12})\cos(\theta_{12}) & -\tan(\varepsilon_{12})\sin(\theta_{12}) & 0 \\ 0 & \sin(\theta_{12}) & -\cos(\theta_{12}) & 1 \end{bmatrix}, \tag{4-53}$$

and $\boldsymbol{b}$ is the angular bias state vector that is being estimated.

*Step 4. Perform the Kalman Filtering Steps to Estimate the Range Bias Parameter.*

At each step $k$ do the following:

$$P^{\Delta r}_{k+1|k} = P^{\Delta r}_{k|k} + s_w^2 \tag{4-54}$$

$$K^{\Delta r}_{k+1} = P^{\Delta r}_{k+1|k} H^T_{k+1} \left[ H_{k+1} P^{\Delta r}_{k+1|k} H^T_{k+1} + \sigma_r^2 \right]^{-1} \tag{4-55}$$

$$\Delta r_{k+1|k+1} = \Delta r_{k|k} + K^{\Delta r}_{k+1} \left[ Z_{\text{range}} - H_{k+1} \Delta r_{k|k} \right] \tag{4-56}$$

$$P^{\Delta r}_{k+1|k+1} = \left[ I - K^{\Delta r}_{k+1} H_{k+1} \right] P^{\Delta r}_{k+1|k}. \tag{4-57}$$

Here, $P^{\Delta r}$ is the covariance of the range bias estimate, $I$ is the identity matrix, $s_w$ is the process noise standard deviation, $K^{\Delta r}$ is the Kalman gain, $H$ is $[1 \;\; 1]^T$, $\sigma_r$ is the standard deviation of the random error in the measurement of the range bias, $Z_{\text{range}}$ is the range bias measurement, and $\Delta r$ is the range bias state that is being estimated.

## 4.8 CORRECTING BIASES

The bias estimation algorithm produces estimates of the angular bias vector $\boldsymbol{b}$ and the range bias $\Delta r$. These quantities are used to correct the second sensor state estimate so it is aligned with the state estimate from the master sensor in position, velocity, and acceleration. The velocity and acceleration alignments are calculated by differentiating equation (3-15). For convenience, rather than using the matrix transformation $\boldsymbol{Cb}$, the position vector is first corrected element by element in spherical coordinates in the simulation, and it is then transformed into Cartesian coordinates. However, the velocity and acceleration corrections are performed in Cartesian coordinates. The following are the alignment equations as used in the simulation.

*Position Correction, in Spherical Coordinates:*

$$r_1 = r_{12} + \Delta r \tag{4-58}$$

$$\theta_1 = \theta_{12} + \Delta\theta - \tan(\varepsilon_{12})\cos(\theta_{12})\Delta\eta - \tan(\varepsilon_{12})\sin(\theta_{12})\Delta\psi \tag{4-59}$$

$$\varepsilon_1 = \varepsilon_{12} + \Delta\varepsilon + \sin(\theta_{12})\Delta\eta - \cos(\theta_{12})\Delta\psi \tag{4-60}$$

*Velocity and Acceleration Corrections, in Cartesian Coordinates:*

$$\boldsymbol{v}_1 = \boldsymbol{v}_{12} + \frac{d\boldsymbol{C}}{dt}\boldsymbol{b} \qquad \boldsymbol{a}_1 = \boldsymbol{a}_{12} + \frac{d^2\boldsymbol{C}}{dt^2}\boldsymbol{b} \tag{4-61}$$

where $\boldsymbol{v}_{12} = \frac{d\boldsymbol{r}_{12}}{dt} = \frac{d\boldsymbol{r}_2}{dt} = \boldsymbol{v}_2$, $\boldsymbol{a}_{12} = \frac{d\boldsymbol{v}_{12}}{dt} = \frac{d\boldsymbol{v}_2}{dt} = \boldsymbol{a}_2$, and the matrix $\boldsymbol{C}$ is presented in equation (3-12). The derivatives of $\boldsymbol{C}$ are straightforward, but lengthy, and are therefore omitted.

## 4.9 TRACK ASSOCIATION AND FUSION

As a measure of effectiveness, a chi-squared test is used to test the statistical *closeness* of the track reported by the master sensor to the corresponding track reported by the second sensor. The test is performed separately for the cases where sensor 2 state estimates are (1) aligned, and (2) unaligned, to the sensor 1 state estimates. The estimates were tested at the 95 percent point, using nine degrees of freedom; i.e., the estimates were associated using position, velocity, and acceleration. The association algorithm uses the difference between the two states and a combined covariance for the two states to compute a statistic $D$ that is chi-square distributed

$$D = (X_2 - X_1)^T (P_1 + P_2)^{-1} (X_2 - X_1), \tag{4-62}$$

where $X_1$ and $X_2$ are the state estimates from sensors 1 and 2, respectively, and $P_1$ and $P_2$ are the covariances of the respective estimates. Note that this formulation of the problem ignores the correlation of the tracking errors from the two sensors for a common target. This was done for simplicity.

If $D$ is less than the chi-square threshold at the 95 percent point for nine degrees of freedom (which is 17.0), the two estimates are assumed to belong to the same target, and the estimates are fused. Otherwise, the estimates belong to different targets and the estimates are not fused. The fusion is computed as follows:

*Compute the Combined Covariance:*

$$P = (P_1^{-1} + P_2^{-1})^{-1} \tag{4-63}$$

*Compute the Fused State:*

$$X = P(P_1^{-1}X_1 + P_2^{-1}X_2) \tag{4-64}$$

# 5.0 COMPARISON STUDIES

Two separate studies were performed using the simulation model for a multisensor system with sensor-level tracking and track-to-track fusion. In this section, these studies will be described and the results of the studies will be presented.

## 5.1 MEASURES OF EFFECTIVENESS

One of the primary measures of effectiveness is the RMSE, computed for the track positions, velocities, and accelerations, before and after alignment. Monte Carlo simulations provide good estimates of the RMSEs, with 100 realizations being performed for each experiment. The percent of successful associations (PA) that occur when the estimates are aligned and unaligned is used as another measure of the effectiveness of the alignment process. Time plots of the bias parameters and various system tracks are used to visualize the convergence of the biases to their true values and the alignment process, as well as to monitor the simulation and to detect inconsistencies in the results.

## 5.2 COMPARISON OF FILTERS AND TIME-ALIGNMENT METHODS

The first study was designed to compare the effectiveness of various tracking filters and time-alignment techniques with respect to bias estimation, alignment, and association. The tracking filters to be compared included a single model filter using a CA model, and a two-model IMM filter with a CV model and a CA model. The time-alignment techniques to be compared were a predictor based on the CA model, a predictor based on the two-model IMM design, a smoother based on the CA model, and a smoother based on the two-model IMM design. Four

configurations of tracking filter/time-alignment techniques were used: CA tracking filters with CA prediction (CA-P), CA tracking filters with both CA prediction and CA smoothing (CA-SM), IMM tracking filters with IMM prediction (IMM-P), and IMM tracking filters with both IMM prediction and IMM smoothing (IMM-SM). For each configuration, 100 Monte Carlo experiments were performed and the RMSEs, PA, and bias convergence plots were generated. The input data and the initial seed for the random number generator were the same for each configuration, so that any differences in the measures of effectiveness could be attributed to the tracking filter/time-alignment configuration.

Both sensors are reporting data with period of 1 sec, but there is a 0.5-sec time lag between the sensors. The measurements from both sensors contain zero-mean Gaussian random errors with standard deviations of 25 m in range and 0.2 deg in bearing and elevation. Both sensors are stationary and the second sensor is located at $t_x = t_y = 25$ m and $t_z = 10$ m relative to the master sensor's stabilized frame. The first sensor has alignment errors given by

$$\begin{aligned} \Delta r_1 &= 25 \text{ m} \qquad \Delta\theta_1 = -0.5 \text{ deg} \qquad \Delta\varepsilon_1 = -0.5 \text{ deg} \\ \Delta\psi_1 &= -0.5 \text{ deg} \qquad \Delta\eta_1 = 0.5 \text{ deg} \qquad \Delta\varphi_1 = 0 \text{ deg} \end{aligned} \tag{5-1}$$

and the second sensor has alignment errors given by

$$\begin{aligned} \Delta r_2 &= 50 \text{ m} \qquad \Delta\theta_2 = 0.5 \text{ deg} \qquad \Delta\varepsilon_2 = 0.5 \text{ deg} \\ \Delta\psi_2 &= 0.5 \text{ deg} \qquad \Delta\eta_2 = -0.5 \text{ deg} \qquad \Delta\varphi_2 = 0 \text{ deg} \end{aligned} \tag{5-2}$$

which gives the following relative alignment errors

$$\begin{aligned} \Delta r &= -25 \text{ m} \qquad \Delta\theta = -1 \text{ deg} \qquad \Delta\varepsilon = -1 \text{ deg} \\ \Delta\psi &= -1 \text{ deg} \qquad \Delta\eta = 1 \text{ deg} \end{aligned} \tag{5-3}$$

The five bias parameters are estimated using two Kalman filters: a first-order filter to estimate the range bias and a fourth-order filter to estimate the angular biases. The covariances of the process noises for the range and angular bias filters are given by $10^{-2}\text{m}^2/\text{sec}^2$ and $10^{-6}\text{rad}^2/\text{sec}^2$, respectively. The inputs to the bias filters are track estimates from the two sensors. The tracking filters at both sensors are implemented with the same parameters, where the process noise covariance $Q_{CV}$ for the CV model and $Q_{CA}$ for the CA model are given by

$$Q_{CV} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \text{m}^2/\text{sec}^4 \qquad Q_{CA} = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{m}^2/\text{sec}^6 \tag{5-4}$$

The initial model probability vector is $\mu_0 = \begin{bmatrix} 0.9 & 0.1 \end{bmatrix}^T$ and the model-switching probability matrix is given by

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \tag{5-5}$$

The target trajectories used in these simulations are presented in Figure 5-1. Two of the targets (targets 1 and 2) are used to estimate the bias parameters, and the third one (i.e., target 3) is corrected using the estimated alignment vector. The alignment was performed dynamically; that is, as bias estimates were generated at a point in time, they were then applied to the track estimates to align them at this point in time. The third target was included to see how well the alignment algorithm performed when other targets are used to generate the alignment vector. For example, in an actual implementation of the alignment algorithm, one or more common targets would be identified for use in the bias estimation algorithm; this report does not deal with the potentially difficult problem of initially identifying common targets before the sensors are aligned. The bias estimates would then be applied to all of the tracks reported by the second sensor to align them to the master sensor. After aligning the tracks, a track-to-track association algorithm could be used to identify common targets reported by the two sensors. Once any common targets are identified, the track estimates reported by the two sensors for a common target could be fused to obtain more accurate state estimates.

The results for the first example are presented in Figures 5-2 to 5-9. In Figures 5-2 to 5-4, the RMSEs (for target 3) in position, velocity, and acceleration, before and after alignment are shown. The mean RMSEs are largest for the CA-P, and smallest for the IMM-SM, both before and after alignment, for position, velocity, and acceleration. The unaligned position errors were on the order of 1200 m, or about 300 percent larger than the aligned position errors. The use of the IMM algorithms for tracking resulted in significantly smaller RMSEs than the CA filters, on the order of 41 m less error for the IMM-P estimates and 78 m less error for the IMM-SM estimates. Since velocity and acceleration are not as sensitive to sensor alignment in sensor-level tracking with track-to-track fusion, the differences between the aligned and unaligned RMSEs for these were not as large, but the improvements due to use of the IMM algorithm and smoothing are apparent.

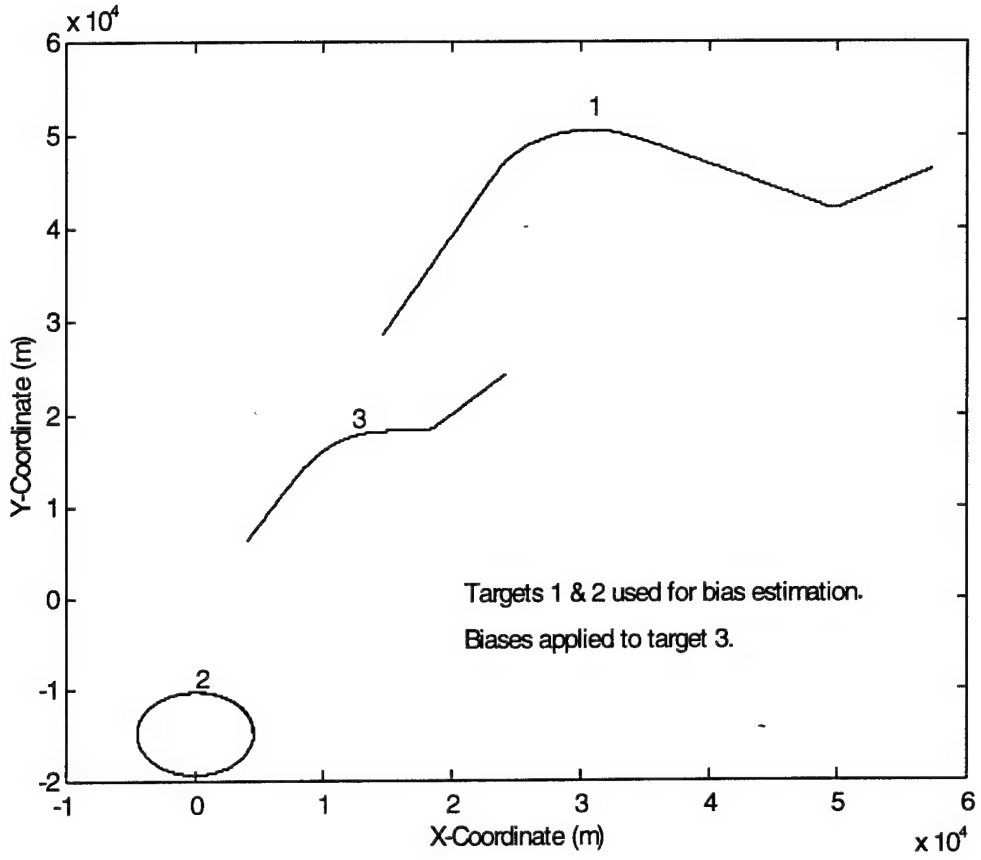


FIGURE 5-1. TRUE TRAJECTORIES OF SIMULATED TARGETS IN XY-PLANE

In Figures 5-5 to 5-9, the convergence of the range and angular bias parameters are illustrated. All the biases converge to nearly their true values in less than 60 sec. The IMM algorithms result in slightly more overshoot in the biases than the CA algorithms, but the RMSEs for the IMM cases are lower after about 30 sec, indicating that good tracking is as important as quick bias convergence to track alignment.

The percent of associated state estimates ranged from 90 to 95 percent for the aligned data and was zero for all unaligned cases. This shows the importance of alignment in multisensor systems. Without alignment, a multisensor system would hold two tracks for each common target; however, with alignment, redundant tracks would occur much less frequently because the common tracks were associated about 90 to 95 percent of the time.

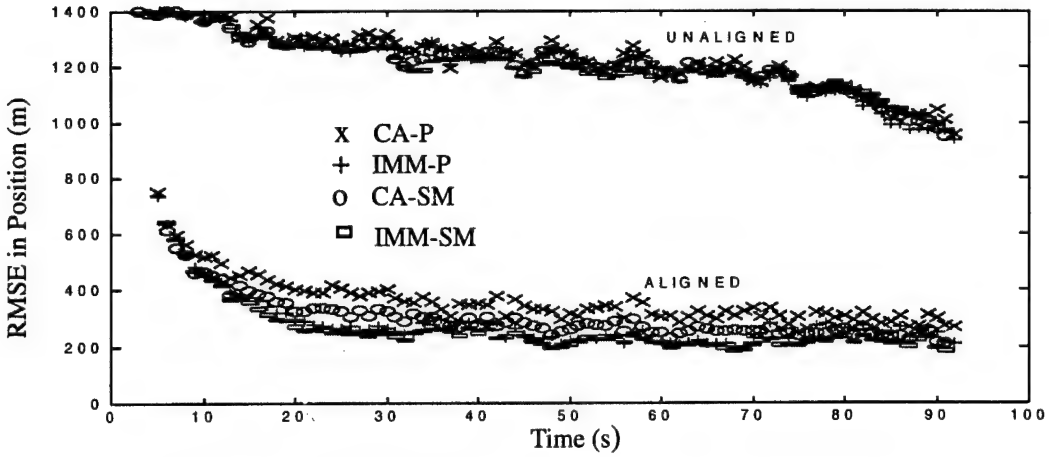


FIGURE 5-2. RMSE IN POSITION FOR FILTER COMPARISON

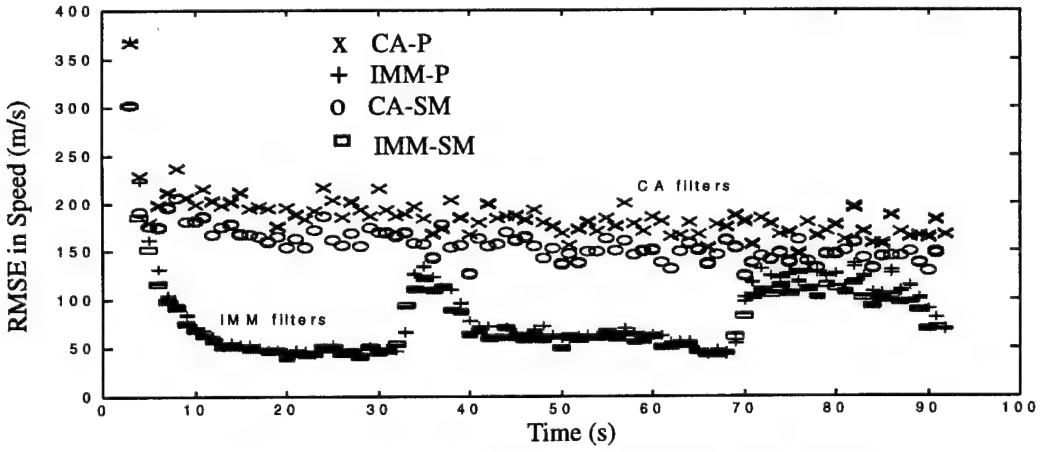


FIGURE 5-3. RMSE IN SPEED FOR FILTER COMPARISON

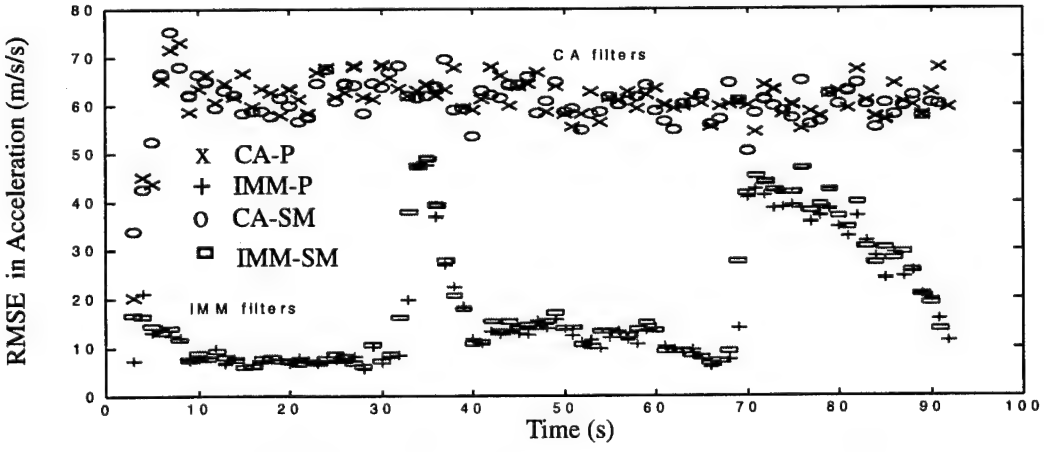


FIGURE 5-4. RMSE IN ACCELERATION FOR FILTER COMPARISON

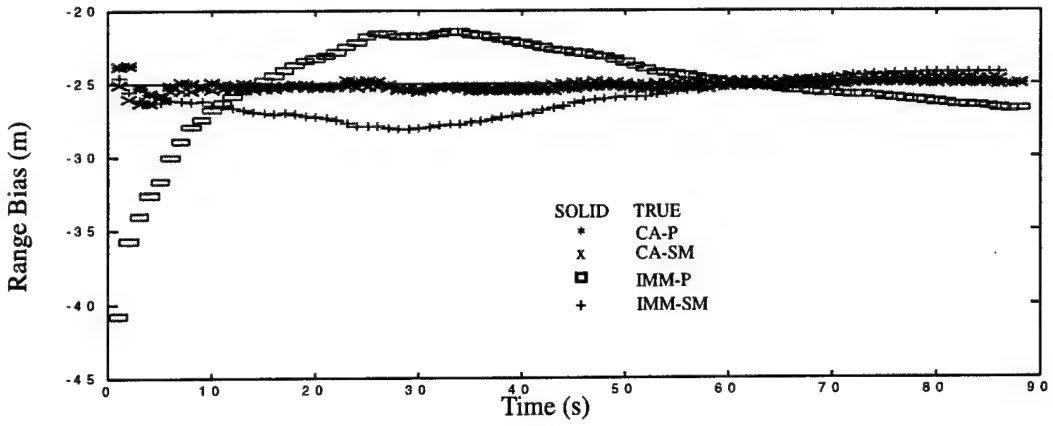


FIGURE 5-5. RANGE BIAS FOR FILTER COMPARISON

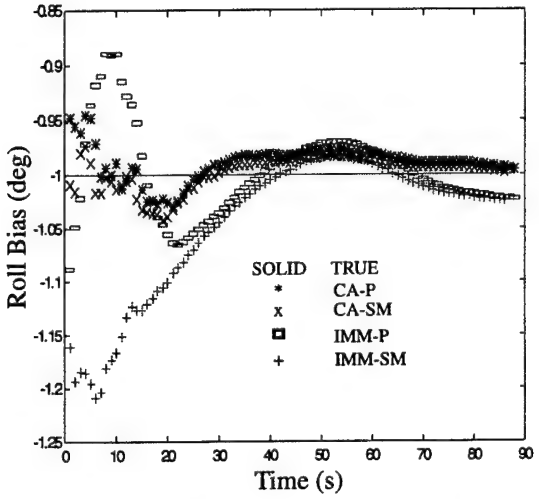


FIGURE 5-6. ROLL BIAS FOR FILTER COMPARISON

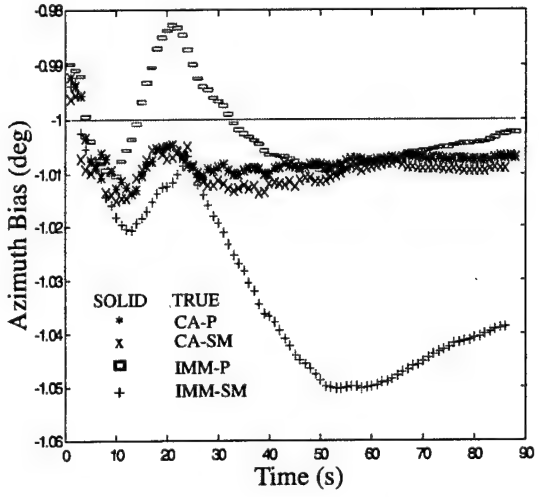


FIGURE 5-7. AZIMUTH BIAS FOR FILTER COMPARISON

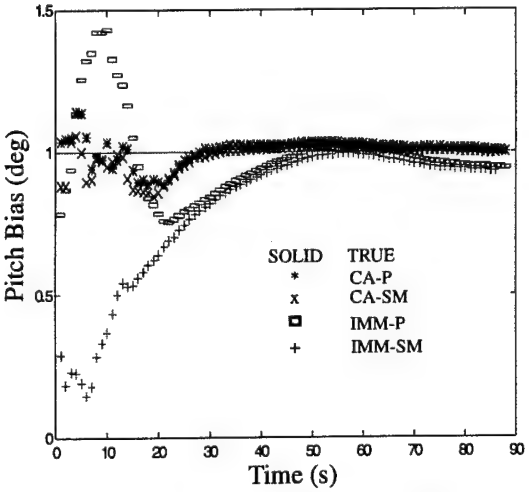


FIGURE 5-8. PITCH BIAS FOR FILTER COMPARISON

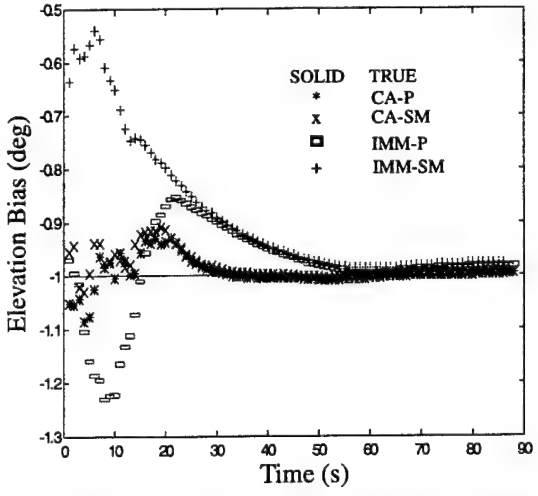


FIGURE 5-9. ELEVATION BIAS FOR FILTER COMPARISON

## 5.3 EFFECT OF BIASES ON ASSOCIATION AND FUSION

The second study was performed to analyze the effects of sensor alignment on track-to-track association and data fusion, using the same simulated sensors as for the first study. There were two phases to this second study. In the first phase, all the biases were set to zero except the bearing bias. The bearing bias was increased in small increments from 0 to 10 deg, and 100 Monte Carlo runs were performed for each azimuth value. The PA was calculated for the aligned and unaligned tracks. The track-to-track association test has a test statistic that is chi-squared distributed with nine degrees of freedom. The 95-percent point for the test statistic is 17; if the test statistic is smaller than 17, the two tracks are accepted as originating from the same target (i.e., the tracks are associated). Figure 5-10 is a plot of the PA versus bearing bias for the aligned and the unaligned tracks. Note that when the sensors are not aligned, the PA rapidly degrades with increasing bearing bias, but that when the sensors are aligned, the PA remains high, between 90 and 95 percent. This reflects the high sensitivity of association to uncorrected sensor biases.

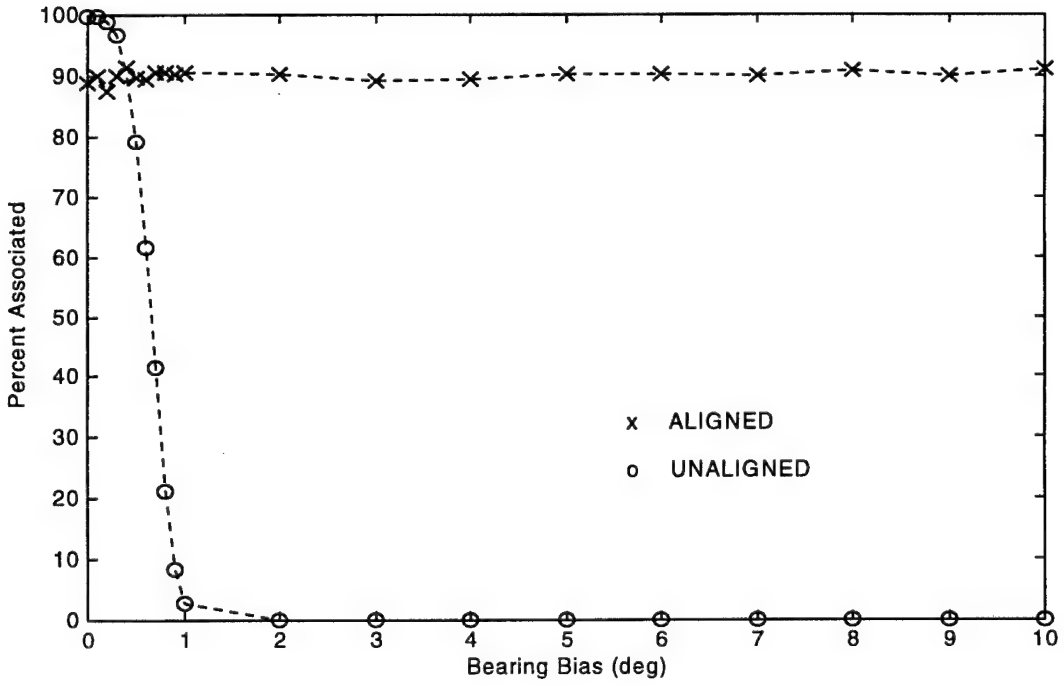


FIGURE 5-10. PERCENT ASSOCIATED VS. BEARING BIAS FOR ALIGNED AND UNALIGNED TRACKS

The second phase of the study was designed to assess the effects of alignment on the efficiency of a generic multisensor track fusion system. One of the sensors was unbiased so that the RMSE reflects that of the second sensor track from the truth, except for random noise. A fused state estimate was formed by assuming perfect association, and statistically weighting and combining the state estimates from the two sensors to form a single state estimate. The RMSE in position for the aligned and unaligned *fused track* was computed for bearing biases of 0 through 0.6 deg in 0.1-deg increments, with 100 Monte Carlo runs being performed for each bearing bias. The RMSE for the *separate tracks* was computed for the zero-bias case only, because the individual sensors are *blind* to their own biases. Figure 5-11 shows a plot of the RMSEs for the unaligned case. Note that the position RMSE of the fused track increases with increasing bias and it is smaller than the RMSE of the individual tracks until about 0.4 deg of bearing bias, above which the error in the fused track is actually greater than that of the error in the individual sensor tracks. In Figure 5-12, the position RMSEs for the aligned fused track are shown. Note that the position RMSE in the fused track always stays well below the error in the individual sensor tracks when the sensors are aligned.

*This study illustrates the importance of alignment to track accuracy and system efficiency. If the fused track were generated by a misaligned system, they may be less accurate than the tracks of the individual sensors.*

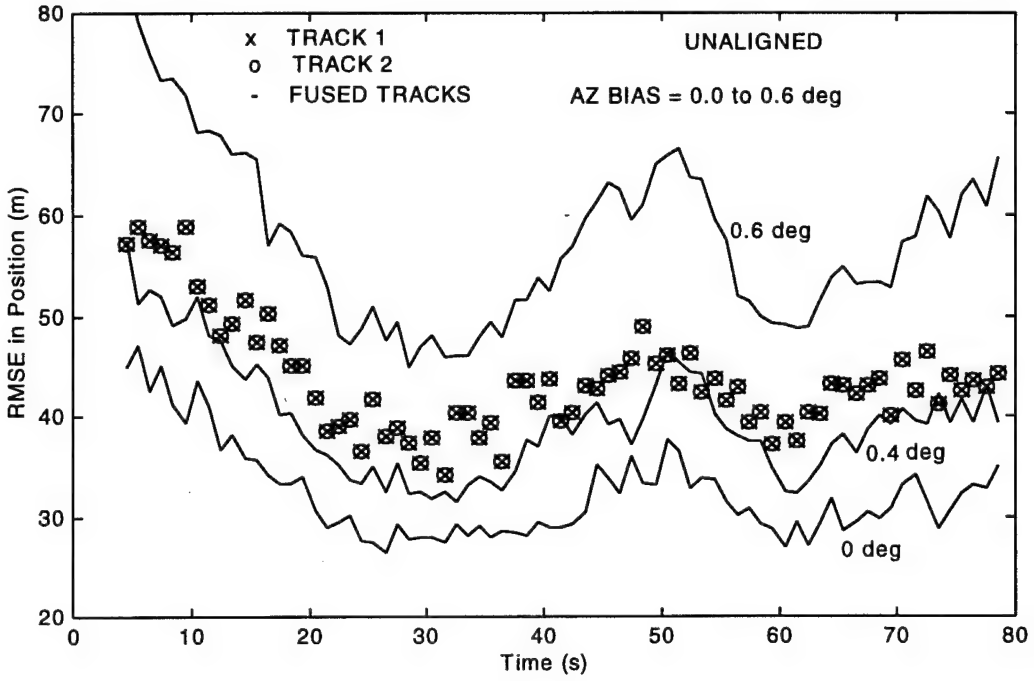


FIGURE 5-11. EFFECT OF INCORRECT ALIGNMENT ON RMSE IN POSITION

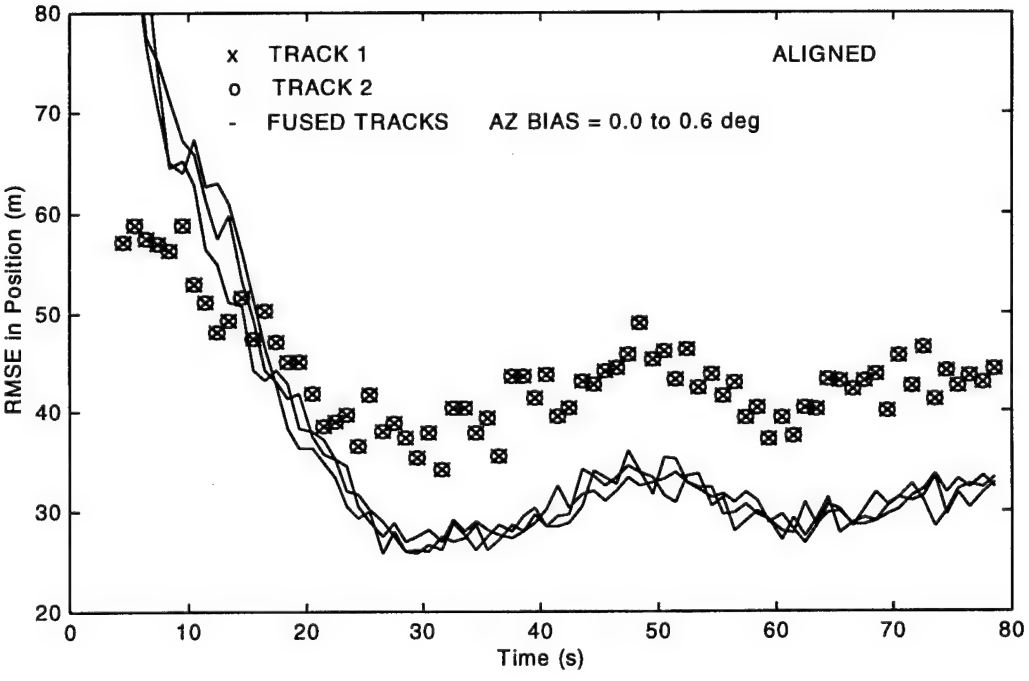


FIGURE 5-12. EFFECT OF CORRECT ALIGNMENT ON RMSE IN POSITION

## 6.0 REAL DATA EXAMPLE

In this section, an example of aligning asynchronous data from a 2D ESM sensor and a 3D radar is described. This experiment was performed under particularly suboptimal data conditions. The sensors were approximately 75 ft apart, but the actual sensor locations are not known. The ESM measured bearing and elevation only, while the radar measured range, bearing, and elevation. A noted difficulty was that the elevation coverage zones of the two sensors only overlapped slightly. This meant that there was only a small chance of observing targets of opportunity that were correlated in both time and space. Also, the number of correlated data points available for such a target was likely to be small. The sensors themselves were not stabilized by means of Inertial Measurement Units, which meant that reported values of roll, pitch, and yaw were not available. Therefore, they were set to zero. A heuristic of assigning the radar range measurement to the 2D ESM measurement was applied, so that the relative range bias was zero. There were only 18 radar data points available to compute the bias parameters. The data was acquired in filtered form, but the exact nature of the tracking algorithm was not known. Thus, only prediction was performed, using the CA predictor. The exact standard deviations in the sensor measurements were not known, but were set to assumed values. Since the radar data rate was lower than the ESM data rate, the 18 radar measurements were predicted to the times of the nearest future 18 ESM measurements. The bias parameters were estimated using the time-aligned radar and ESM data, with the ESM state augmented with the radar range. After allowing the biases to converge, the augmented ESM data were corrected in Cartesian coordinates. Figure 6-1 shows the unaligned ESM, the aligned ESM, and the radar tracks in the XY-plane. In Figure 6-2, the bearing tracks as a function of time are shown. Note that the bearing adjustment was quite successful. The elevation alignment is illustrated in Figure 6-3. This was not as successful because the elevation data for the two sensors was noisier than the bearing data. However, the mean absolute error of the aligned elevation track relative to the predicted radar elevation track was less than that of the unaligned track. The alignment algorithm did improve the elevation estimate, though not dramatically, as it did for the bearing track.

*The results of this real data test indicate that the bias estimation algorithm can permit accurate alignment of dissimilar sensors, and may be able to operate successfully even under highly suboptimal conditions.*

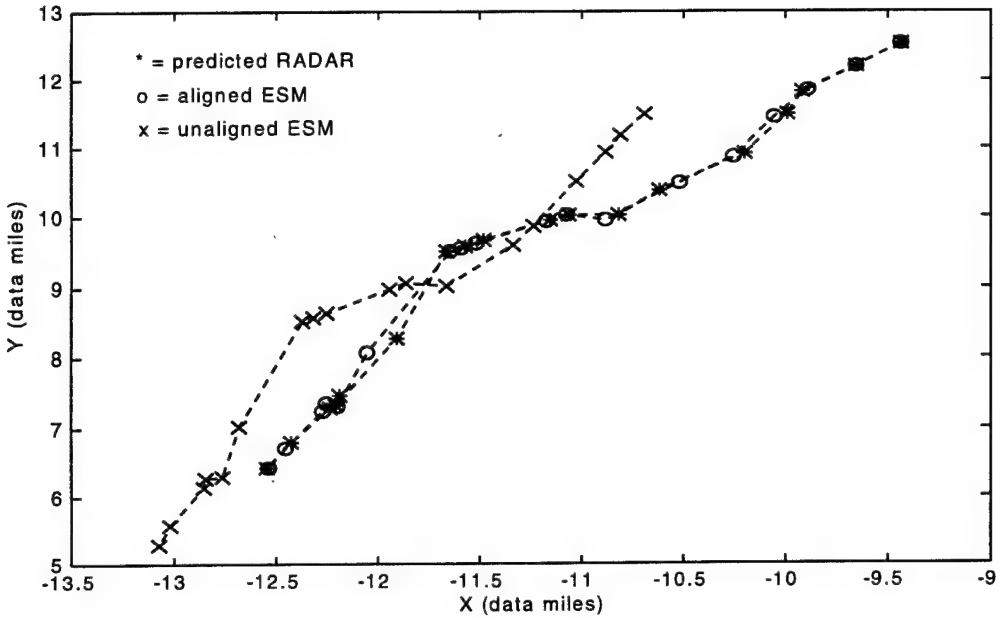


FIGURE 6-1. ALIGNMENT OF RADAR AND ESM TRACKS IN XY-PLANE

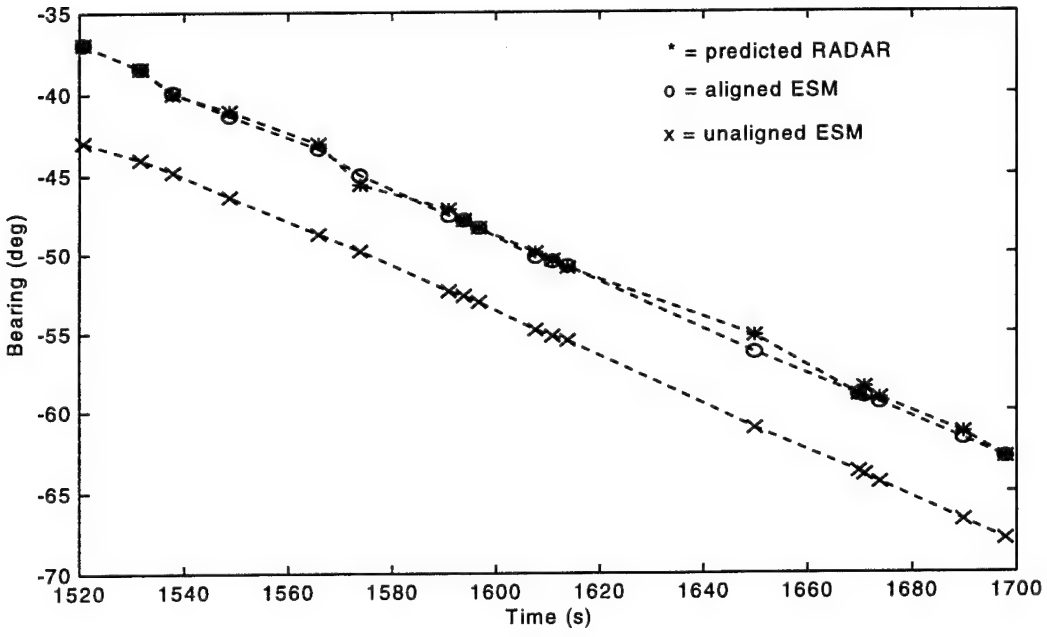


FIGURE 6-2. ALIGNMENT OF RADAR AND ESM IN BEARING

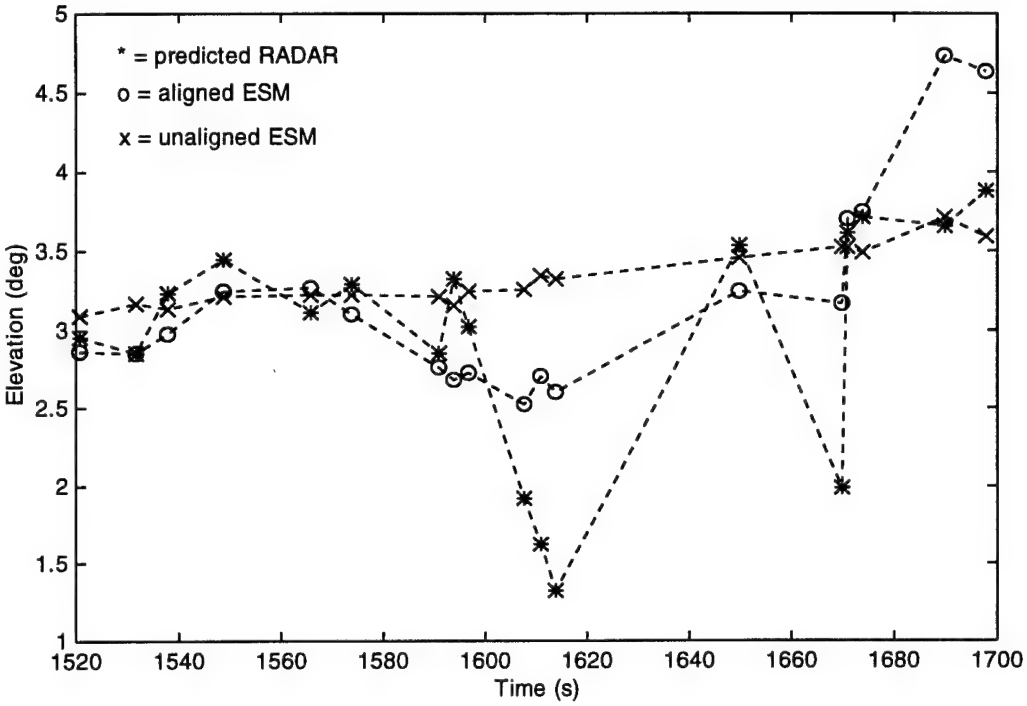


FIGURE 6-3. ALIGNMENT OF RADAR AND ESM IN ELEVATION

# 7.0 SUMMARY

The multisensor alignment problem was examined in this report. Simple simulations were used to demonstrate the effects of alignment errors on multisensor tracking. The results showed that uncompensated alignment errors can seriously degrade the performance of a multisensor tracking system, and they may actually cause the performance to be worse than that obtained using only a single sensor. The amount of degradation in performance depends on the magnitudes of the alignment errors and random noises in the system. For example, it may still be advantageous (i.e., better performance than using only a single sensor) to employ multisensor tracking in the presence of uncompensated alignment errors, if the magnitudes of the alignment errors are of the same order or smaller than the magnitudes of the random noises.

A model was developed to simulate the alignment of two asynchronous sensors in the presence of alignment errors (biases) and random noise. The simulation was based on previous work, as described in references (2), (3), and (5). Included in this study were the use of time-alignment and smoothing algorithms and multiple-model filters. A series of Monte Carlo studies was performed to compare the performances of various filter and time-alignment methods in conjunction with track alignment and association. These studies indicated that track association is extremely sensitive to proper alignment. The studies also indicate that the time alignment of asynchronous data only mildly impacts the amount of error in the aligned track. The same is true for the use of different filtering techniques. Although the differences in performance due to the asynchronous data are small, the IMM filter combined with IMM prediction and IMM smoothing resulted in the smallest aligned track RMSE, while the CA filter with CA prediction only resulted in the largest RMSE. Bias parameters were seen to converge to near their true values in 30 to 60 sec. The state estimates that could be associated with a chi-squared test were between 90 and 95 percent when the tracks were aligned, and was zero when the tracks were unaligned.

A preliminary set of experiments was performed to assess the impact of the time-correlation of the state estimates on the sensor bias estimation problem. The results of those experiments, though not presented in this report, tend to indicate that the time-correlation does not seriously affect the bias estimation or the accuracy of track alignment. However, the results of these experiments are preliminary and more work is needed to ascertain more definite results. It is suspected that the effect of the time-correlation is reduced because the bias *measurements* are *differences* of the state estimates from the two sensors. It would be of interest to investigate this problem both analytically and experimentally.

A real data experiment using filtered data from a 2D ESM and a 3D radar resulted in the successful alignment of the ESM data to that of the radar in the XY-plane and in bearing.

*Based on these results, it is likely that this algorithm would prove to be useful for relatively aligning data from diverse tracking sensors that are stabilized and mounted near to each other, such as those found on a ship. This would increase the efficiency of a multisensor tracking system, lessen the tracking burden by providing more accurate data for track association and fusion, and provide a more accurate picture of the environment.*

## REFERENCES


1. Frieden, D.R., ed., *Principles of Naval Weapons Systems*, Naval Institute Press: Annapolis, MD, 1985.

2. Helmick, R.E., and Rice, T.R., "Removal of Alignment Errors in an Integrated System of Two 3-D Sensors", *IEEE Trans. Aer. Electr. Syst.*, Vol. AES-29, 1993.

3. Helmick, R.E., and Rice, T.R., "Alignment of a 3-D Sensor and a 2-D Sensor Measuring Azimuth and Elevation", Naval Surface Warfare Center Dahlgren Division, NSWCDD/TR-92/181.

4. Helmick, R.E., Hoffman, S.A., and Blair, W.D., "One-Step Fixed-Lag Smoothers for Markovian Switching Systems", *Proc. 1994 American Control Conference*, Baltimore, MD, 29 June - 1 July 1994.

5. Helmick, R.E., Conte, J.E., Hoffman, S.A., and Blair, W.D., "One-Step Fixed-Lag IMM Smoothing for Alignment of Asynchronous Sensors", *Proc. SPIE Aerospace Sensing Symposium-Signal and Data Processing of Small Targets 1994*, SPIE Vol. 2235, Orlando, FL, 4-8 April 1994.

6. Bar-Shalom, Y., and Li, X.R., *Estimation and Tracking: Principles, Techniques, and Software*, Artech House: Boston, 1993.

7. Bar-Shalom, Y., "On the Track-To-Track Correlation Problem," *IEEE Trans. Automatic Contr.*, Vol. AC-26, 1981.

8. Bar-Shalom, Y., and Campo, L., "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," *IEEE Trans. Aer. Electr. Syst.*, Vol. AES-22, 1985.

# DISTRIBUTION

| | COPIES |
|---|---|
| **DOD ACTIVITIES (CONUS)** | |
| ATTN DR RABINDER MADAN 1114SE<br>CHIEF OF NAVAL RESEARCH<br>BALLSTON TOWER 1<br>800 N QUINCY ST<br>ARLINGTON VA 22217-5660 | 1 |
| ATTN E29L (TECHNICAL LIBRARY)<br>COMMANDING OFFICER<br>CSSDD NSWC<br>6703 W HIGHWAY 98<br>PANAMA CITY FL 32407-7001 | 1 |
| DEFENSE TECHNICAL INFORMATION CTR<br>CAMERON STATION<br>ALEXANDRIA VA 22304-6145 | 12 |
| **NON-DOD ACTIVITIES (CONUS)** | |
| ATTN JOHN DI STEFANO<br>VITRO CORP<br>10558 CROSS FOX LANE<br>COLUMBIA MD 21044 | 1 |
| ATTN PROF YAAKOV BAR SHALOM<br>ESE DEPARTMENT U 157<br>260 GLENBROOK RD<br>STORRS CT 06269-3157 | 2 |
| ATTN DR A T ALOUANI<br>DEPARTMENT OF ELECTRICAL<br>ENGINEERING<br>TENNESSEE TECHNOLOGICAL UNIV<br>TTU BOX 05004<br>COOKEVILLE TN 38505 | 1 |
| ATTN EDWARD PRICE<br>FMC CORPORATION<br>1 DANUBE DR<br>KING GEORGE VA 22485 | 1 |
| ATTN MELVIN BELCHER JR<br>MANAGER MISSILE DEFENSE SYSTEMS<br>GEORGIA TECH RESEARCH INST<br>GEORGIA INST OF TECH<br>SENSORS AND ELECTROMAGNETIC<br>APPLICATIONS LAB<br>RADAR SYSTEMS DIVISION<br>ATLANTA GA 30332-0800 | 1 |
| ATTN DR KUO CHU CHANG<br>DEPT OF SYSTEMS ENGINEERING<br>SCHOOL OF INFO TECH AND ENGR<br>GEORGE MASON UNIV<br>FAIRFAX VA 22030-4447 | 1 |
| ATTN SAM BLACKMAN<br>HUGHES AIRCRAFT<br>EO EI MAIL STOP B102<br>PO BOX 902<br>EL SEGUNDO CA 90245 | 1 |
| THE CNA CORPORATION<br>PO BOX 16268<br>ALEXANDRIA VA 22302-0268 | 1 |
| ATTN GIFT AND EXCHANGE DIV<br>LIBRARY OF CONGRESS<br>WASHINGTON DC 20540 | 4 |
| **INTERNAL** | |
| B | 1 |
| B052 (MOORE) | 1 |
| B32 (BLAIR) | 5 |
| B32 (CONTE) | 10 |
| B32 (GENTRY) | 1 |
| B32 (GROVES) | 1 |
| B32 (HELMICK) | 15 |
| B32 (RICE) | 1 |
| B32 (WATSON) | 1 |
| B35 (BAILEY) | 1 |

## DISTRIBUTION (Continued)

| | | COPIES |
|---|---|---|
| B35 | (FENNEMORE) | 1 |
| B35 | (HARTER) | 1 |
| D4 | (LACEY) | 1 |
| E231 | | 3 |
| E282 | (SWANSBURG) | 1 |
| F | | 1 |
| F21 | | 1 |
| F21 | (MCPHILLIPS) | 1 |
| F21 | (SUMNER) | 1 |
| F21 | (MINIUK) | 1 |
| F31 | | 1 |
| F31 | (LYNCH) | 1 |
| F32 | (BOYKIN) | 1 |
| F406 | (HORMAN) | 1 |
| F41 | (KNICELEY) | 1 |
| F41 | (STAPLETON) | 1 |
| F41 | (FONTANA) | 1 |
| F42 | (KLOCHAK) | 1 |
| F44 | | 1 |
| G11 | (LUCAS) | 1 |
| G20 | | 1 |
| G23 | (GRAFF) | 1 |
| N05 | | 1 |
| N05 | (GASTON) | 1 |
| N07 | | 1 |
| N24 | | 1 |
| M24 | (BAILEY) | 1 |
| N24 | (BOYER) | 1 |
| N24 | (BURROW) | 1 |
| N24 | (SERAKOS) | 1 |
| N24 | (MURRAY) | 1 |
| N74 | (GIDEP) | 1 |
| N92 | (MCNATT) | 1 |
| N92 | (LAMBERTSON) | 1 |

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OBM No. 0704-0188*



| 1. AGENCY USE ONLY (*Leave blank*) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | April 1995 | Final |

**4. TITLE AND SUBTITLE**

Real-Time Bias Estimation and Alignment of Two Asynchronous Sensors for Track Association and Fusin

**5. FUNDING NUMBERS**

**6. AUTHOR(s)**

Jeffrey E. Conte and Ronald E. Helmick

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Commander
Naval Surface Warfare Center, Dahlgren Division (Code B32)
17320 Dahlgren Road
Dahlgren, VA 22448-5100

**8. PERFORMING ORGANIZATION REPORT NUMBER**

NSWCDD/TR-94/285

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** ***(Maximum 200 words)***

An extensive simulation study of the problem of relatively aligning two sensors that measure range, bearing, and elevation is described in this report. Simple simulations are used to demonstrate the effects of alignment errors on multisensor tracking. The theory and algorithms for relatively aligning the sensors are briefly summarized. In the original derivation, it was assumed that the data from the two sensors are synchronous. This work presents the extension and simulation of the algorithms to permit the use of asynchronous data. This is accomplished by using Kalman-filter-based prediction algorithms to time-align the state estimates from the two sensors. One-step, fixed-lag smoothing is also employed to improve the accuracy of the state estimates. The effectiveness of using the interactive multiple model algorithm versus single model filtering in the tracking filters prior to bias estimation is also studied. Multiple model versions for prediction and one-step, fixed-lag smoothing of the track estimates are also applied and compared with their single model counterparts with respect to bias estimation accuracy.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| multisensor alignment, track fusion, track association, asynchronous sensors, bias estimation | 68 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and its title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***

**Block 1.** Agency Use Only (Leave blank).

**Block 2.** Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3.** Type of Report and Dates Covered. State whether report is interim, final, etc. *If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4.** Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5.** Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

| | | | |
|---|---|---|---|
| **C** | - Contract | **PR** | - Project |
| **G** | - Grant | **TA** | - Task |
| **PE** | - Program Element | **WU** | - Work Unit Accession No. |

**Block 6.** Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7.** Performing Organization Name(s) and address(es). Self-explanatory.

**Block 8.** Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9.** Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

**Block 10.** Sponsoring/Monitoring Agency Report Number. *(If Known)*

**Block 11.** Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ,,,; To be published in... . When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a.** Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

| | |
|---|---|
| **DOD** | - See DoDD 5230.24, "Distribution Statements on Technical Documents" |
| **DOE** | - See authorities. |
| **NASA** | - See Handbook NHB 2200.2 |
| **NTIS** | - Leave blank |

**Block 12b.** Distribution Code.

| | |
|---|---|
| **DOD** | - Leave blank. |
| **DOE** | - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports. |
| **NASA** | - Leave blank. |
| **NTIS** | - Leave blank. |

**Block 13.** Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

**Block 14.** Subject Terms. Keywords or phrases identifying major subjects in the report.

**Block 15.** Number of Pages. Enter the total number of pages.

**Block 16.** Price Code. Enter appropriate price code (*NTIS only*).

**Block 17-19.** Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of this page.

**Block 20.** Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited or SAR (same as report). An entry in this block is necessary if the abstract is to limited. If blank, the abstract is assumed to be unlimited.